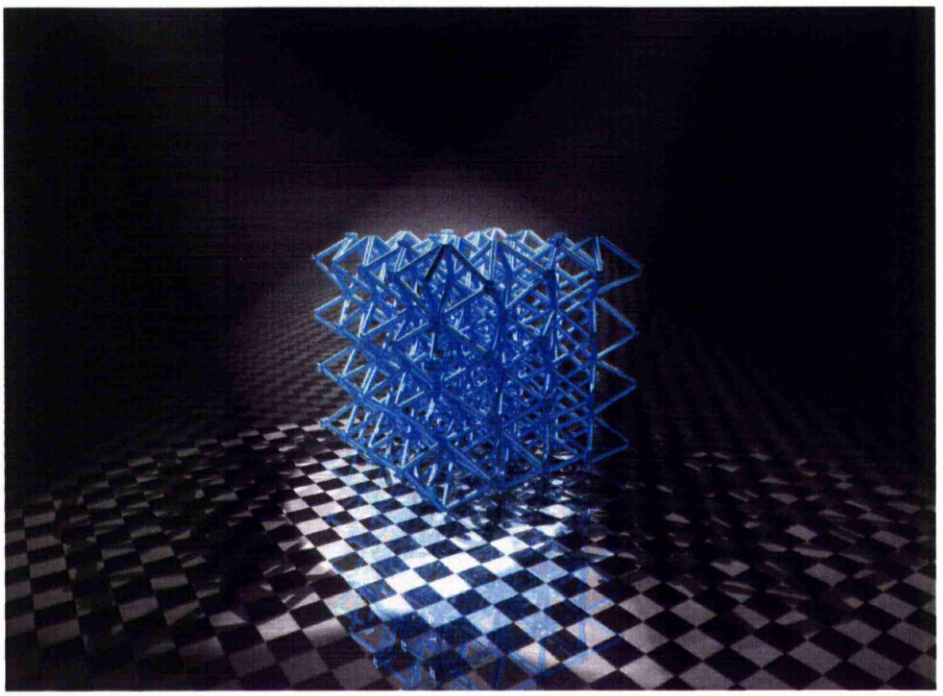
The Creation
of
Lattice Structures
Using
Selective Laser Melting



Thesis submitted in accordance with the requirements
of the University of Liverpool for the degree of
Doctor in Philosophy

Wesley Brooks

March 2011

Acknowledgements

I would like thank my family and friends for the immeasurable support that they have given without request throughout the course of my research. Without this my work would not have been possible.

I acknowledge and am grateful for the support of the academic staff involved during my research. The examiners; Geoff Dearden & Kenny Dalgarno. Supervisory staff; Peter Fox, Wesley Cantwell, Chris Sutcliffe. Experimental officer; Lawrence Bailey.

Finally I dedicate my work to Allan Brooks, my late father who epitomized working beyond expectations.

Signed Declaration

I confirm that other than where explicitly stated or referenced this thesis is my own, original work.

Signed:	(Wesley Brooks)	Date:	(31/03/11)

Abstract

This research focuses on reducing the limitations imposed on the repeating topologies in lattice structures that restrict what can be created using the RealiZer Selective Laser Melting (SLM) machine. The creation of regular, randomly perturbed, polar mapped, and random metallic lattice structures using SLM apparatus is reported and discussed in this thesis. It was observed that a new technique was required to generate the slice data files used to control the SLM equipment in order to create structures that measured significantly more than 10 cells in each axis. The research details the motivations behind the development of the computational methods utilised to develop lattice parts and how the iterations of these methods enabled different areas of research to progress. The limits of the angles from the horizontal that elements could be built are reviewed and scanning techniques are developed that create elements below these values. In order to create horizontal links significant proportions of the machine control software were replaced with software developed during the course of the research. This is discussed at length along with how the limitations on the number of processing parameters available could be removed and how pauses which let sections of the melt on horizontal links freeze before processing the next section could be used.

It is suggested that systems or experimental set ups are developed that allow greater control over the duration of these pauses. This would enable further research into the processing of horizontal links, developing them to the point where they are mechanically consistent and comparable to other links in the structures.

Table of Contents

1 Introduction

Engineering research and practice regularly takes cues from nature. Examples of this are wide spread such as when the early pioneers of flight studied the profiles of various bird wings, through to the use of ceramics to provide durable low friction surfaces, and to the subject of this research thesis, lattice structures. Lattice structures may not be as obvious in nature as the use of ceramics – teeth, bones – and the shape of bird wings but are used in vast quantities inside the skeletal system of all endoskeletal creatures. Figure 1 Shows an example of a lattice structure which is present in the volume of the shown bovine femur that is placed under multi-directional stress. The structure is an example of a hierarchical sandwich structure, with a lattice core provided by the cancellous bone and the compact bone providing the skin.

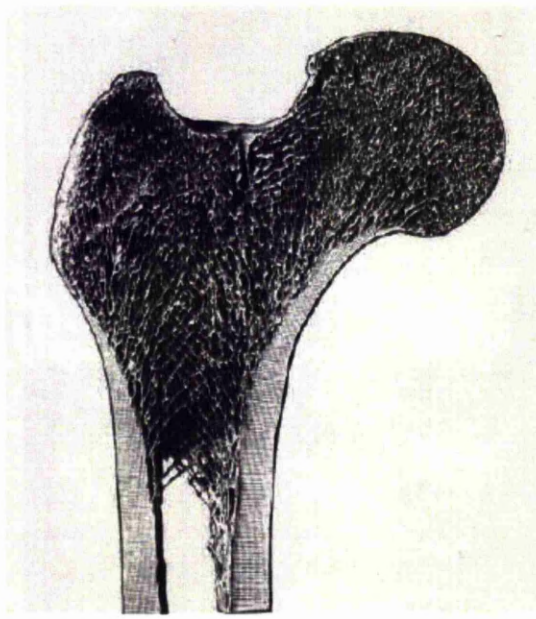


Figure 1: Bovine femur showing a natural example of lattice structures, cancellous bone [1].

Nature has been driven to the use of these structures to create high performance

materials that are as efficient as possible. There are many driving forces that have led modern engineering to increasingly investigate the use of lattice structures. One example is the adoption of lighter weight components in aerospace and automotive industries to reduce global high energy consumptions and the side effects of many energy production methods. Lighter components that perform equally well or out perform their predecessors [2], would be of interest to other sectors of engineering which are ultimately driven by reaching increasing performance targets such as in the development of modern high performance cars, trains and aircraft [3].

Lattice structures enable the production of light stiff systems where the weight of the base material of the lattice is placed where it is used best. The ultimate aim is where an engineer can design a component using a lattice structure that places mass only where it is needed, while having to work around as few limitations as possible due to the manufacturing process.

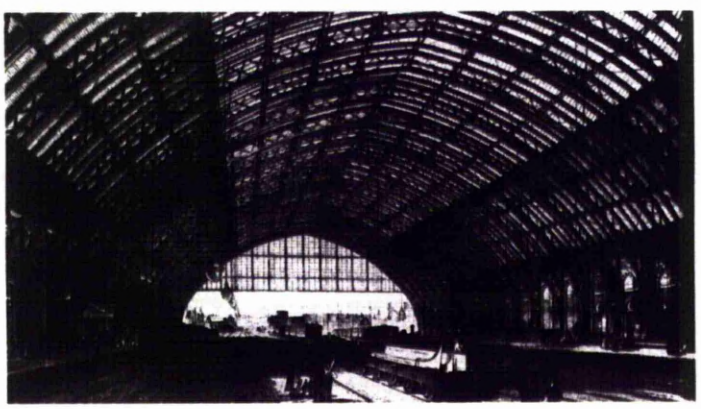


Figure 2: St Pancras Station in London while under original construction.

The use of lattice structures in engineering is not new. Examples are present in roof structures with large spans such as those in the Victorian train stations Liverpool Lime Street and St Pancras station. The latter of which is shown in

Figure 2 whist under construction. What is different to the focus of this research is that the interest is in developing a method for the creation of lattice structures with higher complexity at far smaller length scales, millimetres and below, rather than metres.

Cellular metal materials have been in development since the 1940s, and the level of research has increased vastly over the last three decades [4]. There are many ways to produce a metallic foam or truss structure with various levels of porosity, in various materials, cell sizes, and there are a few methods available to create designed cell topology. One of the current problems is that most techniques that allow some degree of topology design, such as woven textiles, assembled pressed sheet and mesh structures have a great deal of limitations on the geometry of the topology, or require many stages in the production of the lattice component. Designed lattice structures are particularly suited for use in cored sandwich structures which are loaded in compression or bending. The octet truss style of topology has been highlighted as of particular interest due to the loading of all trusses being tension or compression with no bending [2], [5]. There are very few options available for the manufacture of these designed structures (as will be demonstrated in the literature review) and techniques suitable for the manufacture of a core with unit cell sizes around 2-5mm (below the minimum sizes for cast parts) are very limited, particularly so if there is more than one layer of cells.

1.1 Applications for Metallic Lattice Structures

The use of lattice structures in engineered components have many advantages over currently utilised techniques.

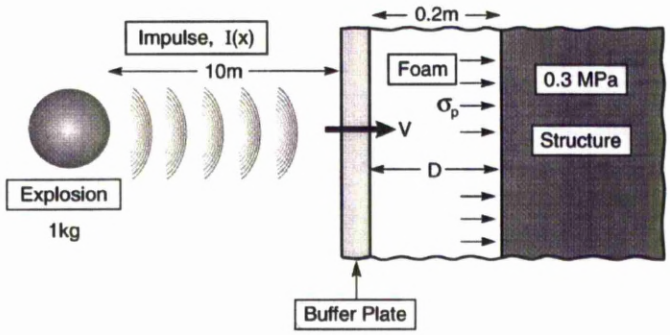


Figure 3: Use of foams for blast protection [6].

As the structures can be designed so they have large porosity and potentially a designed failure mode due to the arrangement of the trusses the structure can be used in impact absorption where impact energy would be absorbed by the progressive plastic deformation of the lattice. This is demonstrated by the foam structure in Figure 3. Metallic closed cell foam structures have been used for impact [7] and sound absorption [8], however to gain an optimum performance from the raw foamed material significant additional processes such as drilling and rolling have to be carried out.

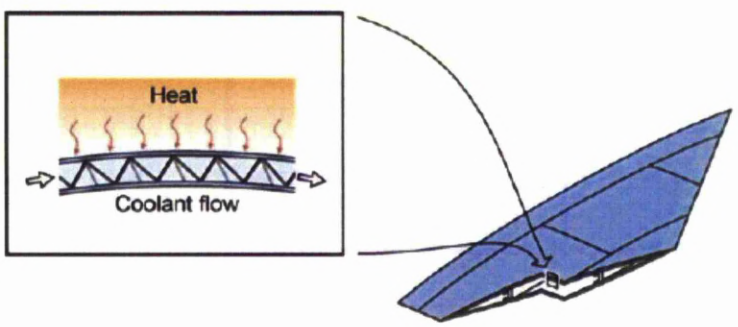


Figure 4: Multifunctional lattice structure in a wing skin. Cooling and load bearing. [9]

Cellular materials can be used in load bearing structures. Where the manufacturing technique permits the truss elements of the lattice structures can be aligned to support the applied load in an efficient manner. When the lattice

element can be made with a small radius and high aspect ratio the surface area to internal volume ratio is high, so they lend themselves to heat transfer applications. Figure 4 shows the use of a lattice structure as a multifunctional material. The lattice is load bearing, and providing a large surface area for heat transfer [10], conducted from the surface of the wing, to the coolant through the lattice. The complex shapes achievable with lattice structures can also be of benefit to the development of smart structures. [11], [12].

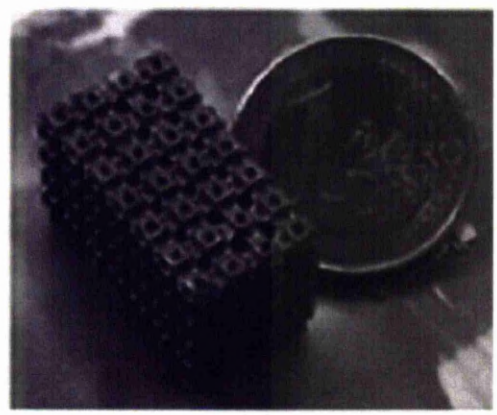


Figure 5: Titanium cage built to test the feasibility of the use of SLM parts for use in Spinal Arthrodesis. [13]

Lattice structures have application within the natural systems that inspired their use in the first instance. The example in Figure 5 shows a component which is being developed as part of a spinal operation to immobilise adjacent vertebrae where other surgical methods are not available nor is it possible to recover the use of the joint. Other authors have proposed the use of lattice structures produced by layer manufacturing for use in joint replacements where the lattice is used both to reduce stress shielding, and as a scaffold for bone ingrowth.

1.2 Layer Manufacturing

Layer manufacturing (LM) covers a range of manufacturing techniques that

produce three dimensional components from a succession of layers. The process of creating components from layers, rather than moulding or cutting back from billets of materials enables the creation of otherwise impossible geometries.

Generic limitations of layer manufacturing include a visual stair-stepping effect on surfaces that are a low angle from the layer plane, and minimum feature definition in the direction of the layering limited to one layer thickness. Further restrictions are implied by the precise technique, material, and processing conditions being used and can vary from shrinkage and offsets, minimum feature definition, through to minimum build angle from the horizontal and supporting requirements.

Figure 6 details a cube that is being successively sliced at a regular layer thickness and highlights a single slice layer and intersection half way through.

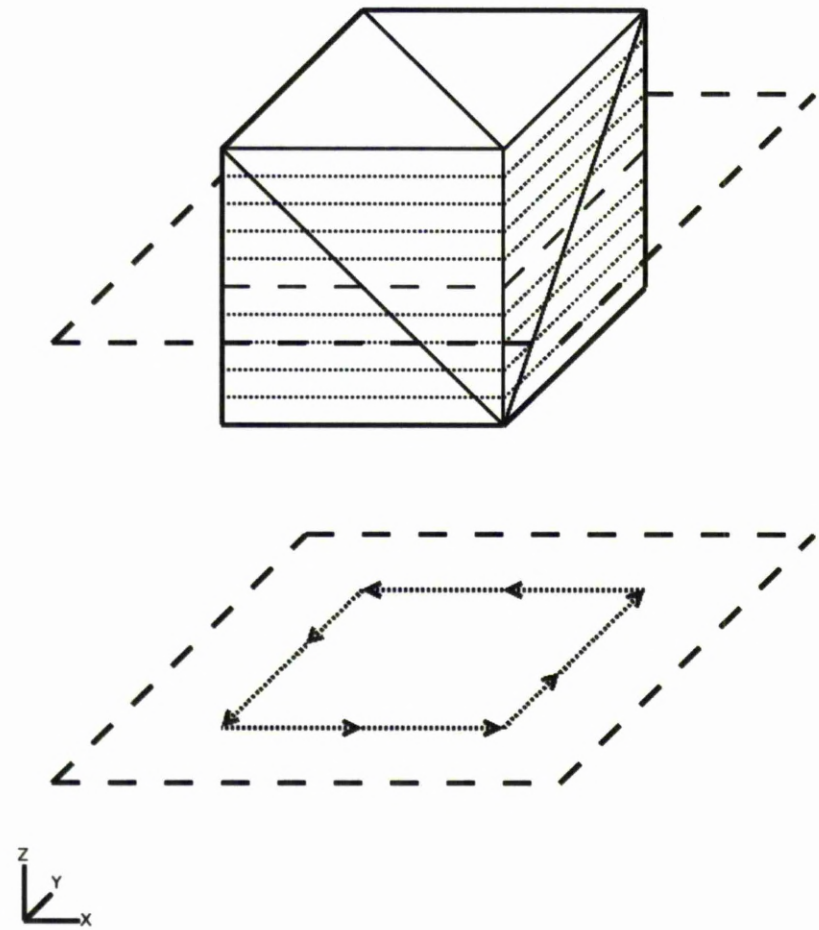


Figure 6: Detailing the slicing of a simple cube stl file and the resulting contour.

As well as defining the bounds directed energy layer manufacturing techniques require information on where to direct the energy that is required to either activate a chemical reaction to set a liquid, or provide thermal energy to fuse or fully melt particles of material. The contour extracted from the cube being sliced in Figure 6 is further processed in Figure 7. Here the extracted contour is detailed as the original outer contour. In order to create a solid cube the raster hatch is required to fill the area. The contour path is a finishing operation used to improve the outer surface finish of components. The hatch spacing, offset between the hatch and the original outer contour, and the new contour and the original is to account for the

dimension of the energy beam being delivered to the bed and the dimensions of the proportion of material that it effects.

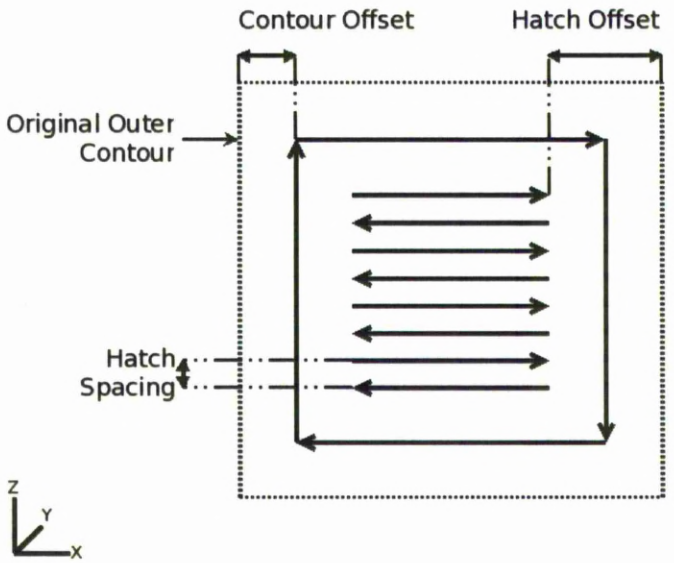


Figure 7: Detailing the generation of the scan vectors for the slice generated in figure 6.

Being able to define where the solid is on a layer by layer basis lends itself to creating highly complex lattice structures. The generic limitations imposed by LM do not prevent the creation of complex lattice structures. The LM technique utilised in this research (briefly detailed next and fully detailed in the methodology section) is in it's infancy, and so while further limitations are known these can be reduced by focused research.

1.3 Direct Metal Processes

During the initial stages of the research there were a number of commercially available additive manufacturing processes that were capable of directly creating metal components without the requirement for post processing steps such as binder burn out and infiltration. These have been identified in Table 1.

Company	System	Process	Energy Source	Power
Arcam AB	EBM	EBM	Electron Beam	1kW+
Concept Laser	LaserCURSING	SLM	Ytterbium Fibre	200W
EOS	EOSINT M 250/270	SLM	Ytterbium Fibre	200W
MCP	SLM Realizer	SLM	Ytterbium Fibre	400W
Optomec	LENS	Laser Cladding	Nd:YAG/Fibre	500W - 2kW
Phenix Systems	PM250	SLM	Ytterbium Fibre	200W
POM	DMD	Laser Cladding	CO_2	6kW

Table 1: Direct metal additive manufacturing processes [14]

As identified there are three main processes, Electron Beam Melting (EBM), Selective Laser Melting (SLM), and Laser Cladding. These are briefly discussed in the following sections.

1.3.1 Electron Beam

As identified in Table 1 the only available EMB system was supplied by Arcam AB. The process as detailed in Figure 8 is carried out in a vacuum and once the chamber is prepared (heated and under vacuum) comprises of 3 main stages. The first assuming a pre placed powder bed from the last cycle is a preheating operation that heats up the top layer of powder local to the area to be scanned and partially sintering it. The second is a melting stage where the electron beam is melting a selected area of the powder as described by the scan data. The final stage (split into two in the figure) is the add layer process where the elevator dips down one layer thickness and a new layer of powder is added. These three steps are repeated until the build is complete.

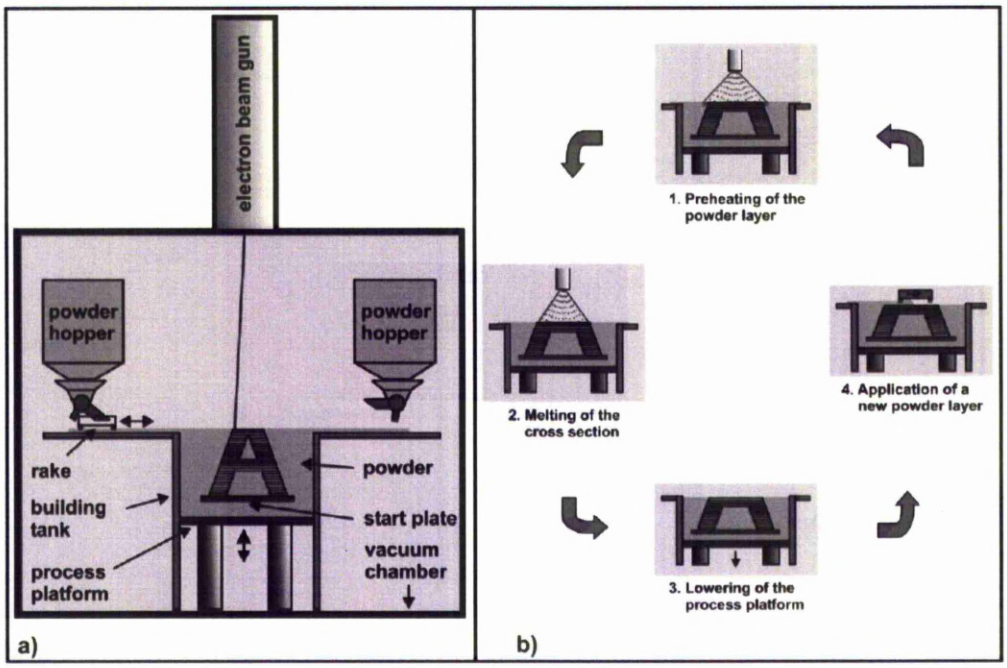


Figure 8: Schematic of the Arcam EBM 12 system (a) and component generation layer by layer (b). [15]

1.3.2 Laser Cladding

Unlike SLM and EBM Laser Cladding does not rely on a pre-placed powder bed. The required powder is sprayed into the melt site within a protective gas. While not all the powder sprayed is included into the melt pool it does give the flexibility of allowing multiple materials. Figure 9 details the spraying of the powder into a melt pool and stacking a few melt tracks on top of each other – the basis of the part manufacturing process. The spraying and laser aperture are moved around either by a CNC head, or the nozzles remain static while the work piece is moved around beneath it.

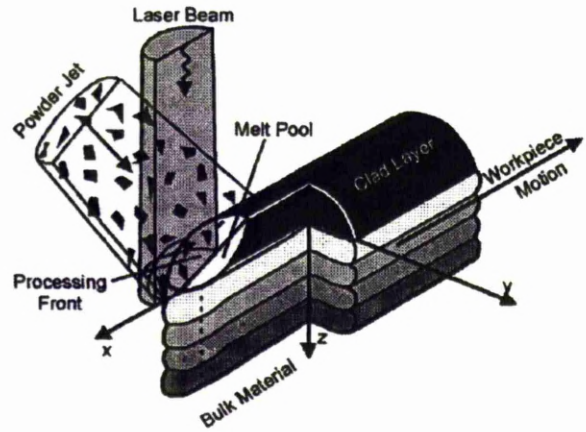


Figure 9: Process schematic of the laser cladding process. [16]

Figure 10 Details the process control of the lens system and also details three powder hoppers, meaning a part can be created with a constantly changing alloy mixture of three base metals.

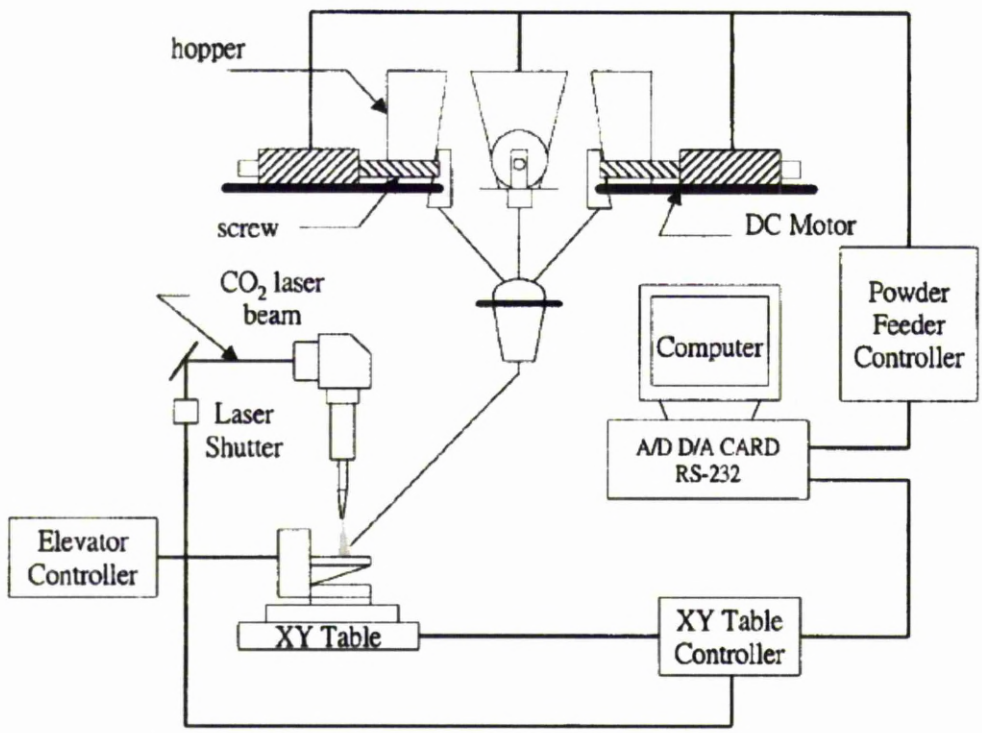


Figure 10: Control schematic of laser cladding. [17]

1.3.3 Selective Laser Melting

SLM provides advantages over current manufacturing methods for lattice structures (as detailed later in the literature survey) in that it is largely a single step process to manufacture single material components, and the cell topology can be designed, so long as the designed topology is within the limitations of the SLM process. The machine used in this research is shown in Figure 11.

Figure 11: MTT SLM Realizer II

Figure 12 Is the same machine with the doors open showing the locations of some of the major components. Note the use of an SPI laser system rather than IPG system that is more common on the Realizer machines.

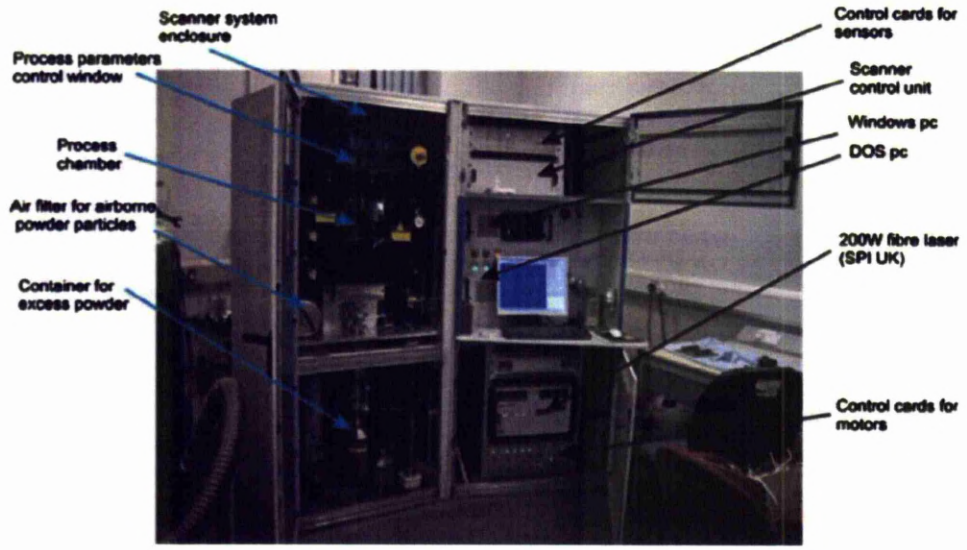


Figure 12: Details of the SLM Realizer II [18]

1.4 Introduction Summary

The interest in lattice technologies has clearly risen in recent decades. The basic idea of using lattice structures to make most effective use of material in order to minimise weight while maintaining structures is much older as shown by the extensive use of assembled lattice structures in bridges and large span roof structures in the Victorian age. As research and manufacturing techniques have developed it has become feasible to create lattice structures whose cell geometries are small enough to be usable in aircraft and impact absorption structures. Methods to create the lattice structures with designed cell structures, at a much reduced length scale required further development. Direct metal layer manufacturing techniques give the ability of free-form creating structures out of any material that the laser can melt. There are however challenges to overcome in order to further develop the knowledge of what is possible with these machines, specifically the SLM Realizer machine.

2 Aims for Research

The SLM technology has limitations that are generic to LM and also specific limitations that are particular to SLM. The current understanding of the limitations significantly limits the scope for creating lattice structures using the SLM technique. These limitations are identified when they are encountered during the research and reduced as far as possible in order to expand the scope of exploitation for lattice structures created on SLM machines.

In order to increase the scope of applications for the created lattice structures any limitations to the complexity and minimum cell size need to be identified and methods to reduce these researched. File sizes and low link angles are areas of initial concern.

Any software tools created during the scope of the research need to be available to, and usable by researchers working on parallel collaborative projects that are investigating applications for metallic lattice structures in heat exchange and medical applications.

In order to increase the scope of application for the lattice structures as far as possible the topology of the created lattice structure must be variable. This will enable the structures to be tailored to best suit the chosen application. Cell topology can be varied by cell size, changing the relative size in x, y, and z axis, and changing its geometry.

3 Literature Survey

3.1 Manufacturing Methods for Metallic Lattice Structures

3.1.1 Metallic Foam Structures

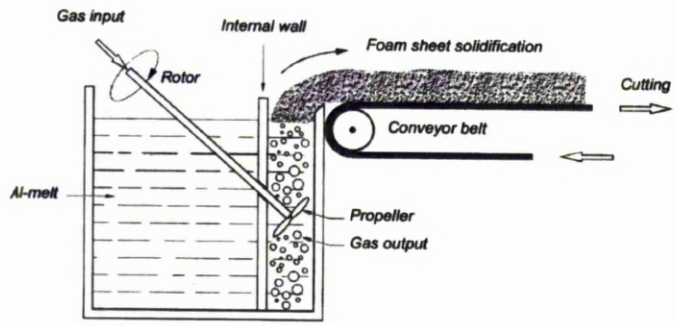


Figure 13: Direct foaming by gas injection. [4]

Metallic foam materials can be, and are commercially produced by directly foaming the molten material and freezing the melt before the bubbles escape. The approach detailed in Figure 13 shows the gas being injected in to the melt coaxially to a rotating propeller shaft. Ceramic particles are added to the melt to increase the viscosity and hence reduce the speed at which the bubbles rise [4], [19], [20]. There are two challenges in this process; gaining a consistent and desired pore distribution, and freezing the melt before the bubbles escape. As the metal melt is very dense the gas tends to rise through the melt very quickly due to the large buoyancy forces. The propellers main functions are to distribute the ceramic particles and bubbles, and reduce the individual bubble size to a consistent level.

The porosity of these materials when made in aluminium is 80-98% with cell sizes ranging from 30 to 3 millimetres [19] and wall thickness from 50 – 85 microns. Materials that foams can be made from include aluminium alloys; casting grade A359, wrought grades 1060, 3003, 6016 and 6061. These foams are

produced on a commercial basis by Hydro Aluminium in Norway, and Cymat Aluminium in Canada. The latter produces the foam in quantities of up to 1 metric tonne per hour in blocks measuring 1.5m wide, and 250mm thick. The disadvantages of this product are that the porosity of the foam can vary through its height as the bubbles move in the foam up to the point at which it freezes on the conveyor and the pore shape can be affected by the shear forces introduced in the foam as it is pulled from the top of the melt. The particles added to stabilise the foam also cause the final foam to be more brittle. [4].

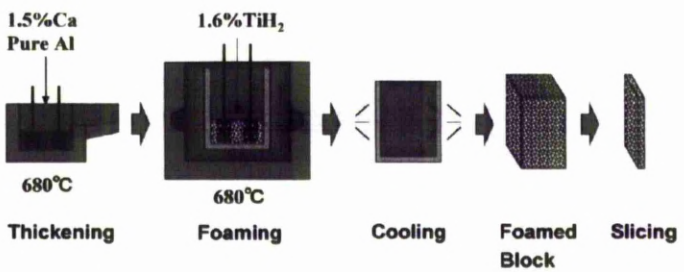


Figure 14: Direct foaming by blowing agent.[4]

Another direct melt foaming technique uses the decomposition of chemicals to releasing gas in the melt as shown in Figure 14. Calcium is added into the pure aluminium melt. This thickens the melt, up to five times it's original viscosity. Following this titanium hydride is added which decomposes and causes the melt to slowly expand as it decomposes and produces gas. At this point the melt is allowed to cool in a mould where the pores continue to form and expand. This method produces the most homogeneous distribution of pores from a direct foaming process [4]. Porosity for the material can range from 80-93% with cell sizes of 0.5 to 5 millimetres [21]. Simone et al report a larger cell size range of 2 – 20mm [19].

This has been commercialised on a small scale by Shimo Wire of Japan [22]. The production volume is approximately 1000kg per day in blocks measuring 2.05 x 0.65 x 0.65m.

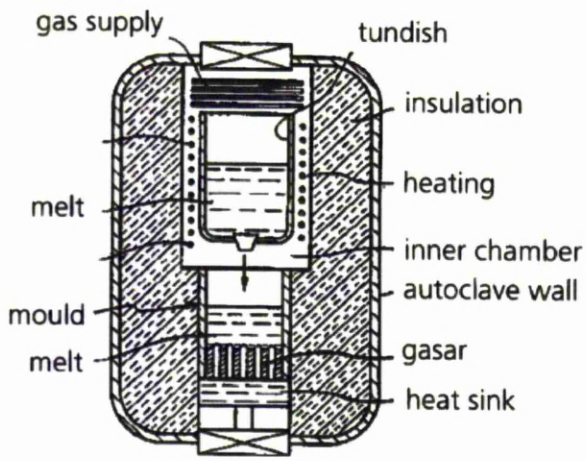


Figure 15: A production method for gasar foams. [4]

The term "gasar" comes from a Russian term, which translates as "gas-reinforced". When some metals are melted under a high pressure in a hydrogen atmosphere the hydrogen is absorbed into the melt to form a single phase system. When cooled the melt returns to a two phase system of solid metal and hydrogen. If this cooling process is completed under a controlled environment as shown in Figure 15 then long cylindrical pores are formed, aligned with the direction of cooling. The pore diameters can range form 10 microns to 10 millimetres with lengths of 100 microns to 300 millimetres. Overall porosity varies from 5 to 75% [4].

Radial and axial pore geometries can be formed depending on how the melt is cooled. Nickel, copper, aluminium, magnesiums, steels, cobalt, chromium, molybdium and ceramics have all been processed with this technique.

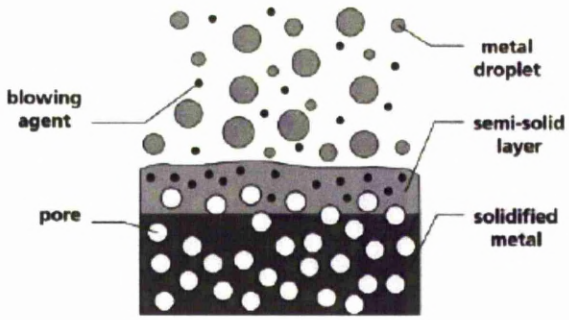


Figure 16: Manufacture of cellular material by spray forming with a blowing agent. [4]

Closed cell foams can also be manufactured by spray forming methods as shown in Figure 16. This method uses a blowing agent to create pores in the melt as it solidifies. The difference between this and the process shown in Figure 14 is that the melt and blowing agent are sprayed onto a substrate, rather than the blowing agent being added into a volume of molten metal. A wide range of metals can be processed in this way, and porosity can reach 60%, but the distribution of pores and their sizes are not uniform. Bronze and steel foams have been produced using this technique. [4], as well as aluminium alloys [23].

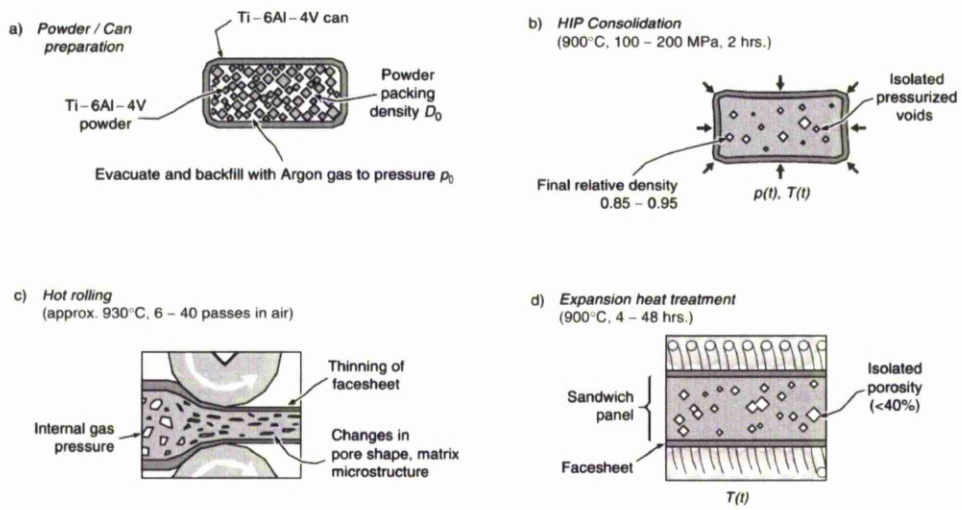


Figure 17: Manufacturing method for entrapped gas closed cell material. [21]

The procedure for manufacturing entrapped gas cellular materials is detailed in Figure 17. Step a shows the powdered metal contained in a canister made of the same material which is subsequently evacuated and back filled with argon. Argon is used as there is a very low tendency for the argon to be absorbed into the metal. Step b shows the sealed canister undergoing hot isostatic pressing, where the overall porosity of the material is reduced to 2-10%. The gas trapped in the voids is now under high pressure. A hot cross-rolling stage shown in step c improves the distribution of the pores within the material. The final stage heats up the metal until the point where the gas pressure is great enough to expand the metal through creep. The resulting part porosity is around 50%.[21] This process is used by Boeing to produce porous titanium components [4].

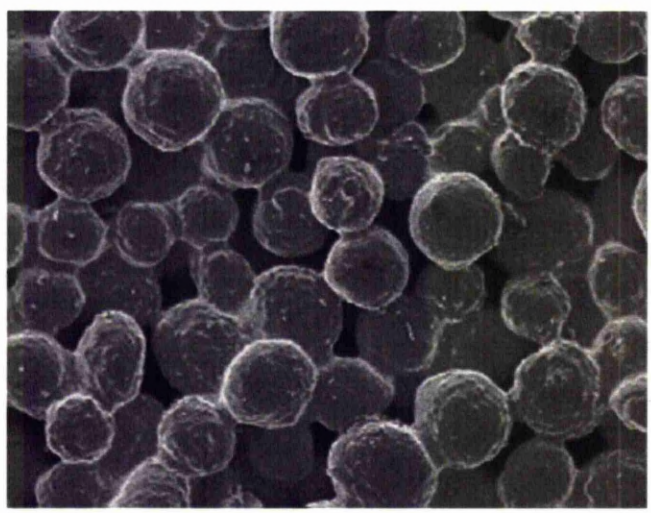


Figure 18: Porous sintered bronze from particles with roughly 100 micron diameter. [4]

A powder sintering technique is shown in Figure 18. The material shown is made by loose or gravity packing the powder then sintering. This technique yields porosities of between 20 and 50%, but creates a relatively weak structure. Aluminium powder is difficult to process using this technique as the particles are

normally covered with an oxide layer. To process aluminium the powder must be deformed to break the oxide, or sintering aids are needed. These aids – such as copper, silicon, or magnesium form a eutectic alloy. A eutectic alloy is where the added sintering aid is mixed with the aluminium to form an alloy with a lower melting point. Eutectic is derived from a greek word literally meaning easily melted.

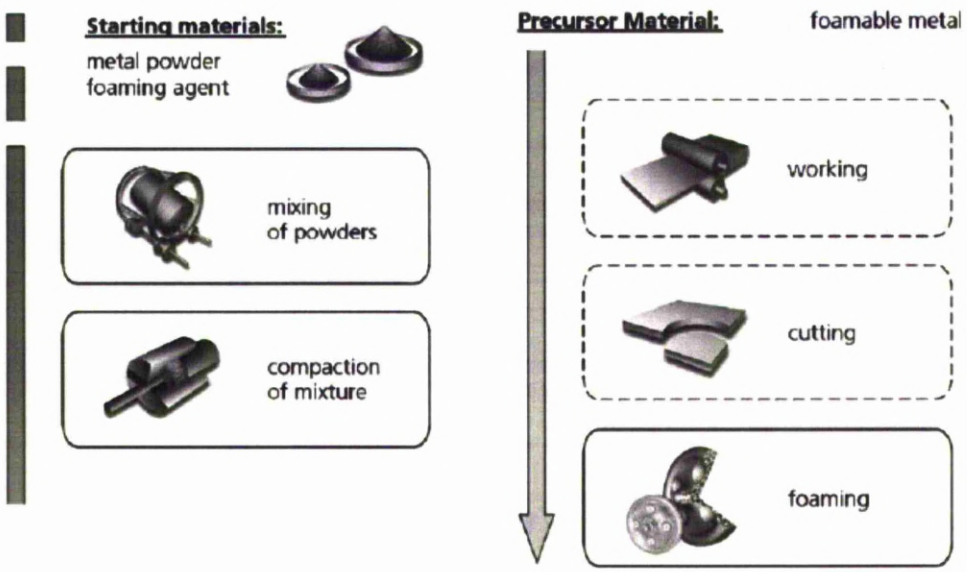


Figure 19: Powder compact melting process. [4]

Another powder metallurgy method is presented in Figure 19. In this example the powder is mixed with a blowing agent while both are in a powder form. After mixing the powder is compressed and worked into a solid billet with as little porosity as possible. A section of the billet can is then raised in temperature which causes the powder to sinter and the blowing agent to decompose and cause the material to expand slowly. If this is done inside a mould it will take the shape of the container. Steels, aluminium, zinc, titanium, brass, lead, and gold can all be processed with this method. Aluminium alloys in the grades 2xxx, 6xxx, and

A356 are most commonly used.

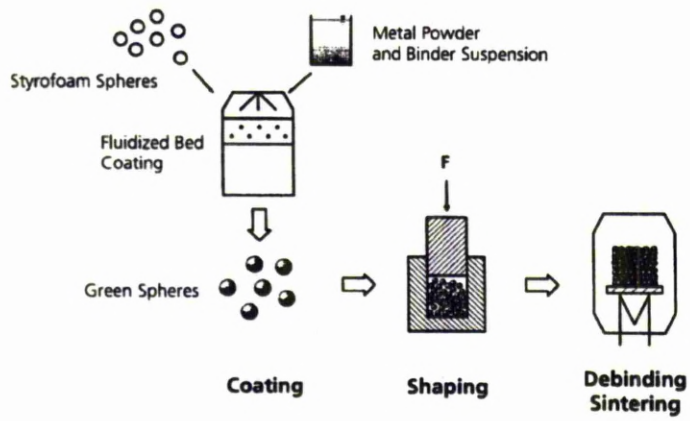


Figure 20: Styrofoam hollow spheres manufacturing technique. [4]

Figure 20 shows a method using powder sintering Styrofoam spheres. The spheres are coated in a powder and binder material. These spheres are then placed into a shaping mould. They may or may not be ordered and can be compacted at this stage to reduce the open porosity. Once in the mould the de-binding and sintering of the metal powder is carried out in the same step. The resulting structure can contain both open and closed pores and has an overall porosity of approximately 80%. 36% of the porosity is open pore, and 44% closed volume within the space occupied by the spheres. The exact values and ratio of porosity depends on the packing, size distribution, and thickness of the binder/powder mix applied to the spheres.

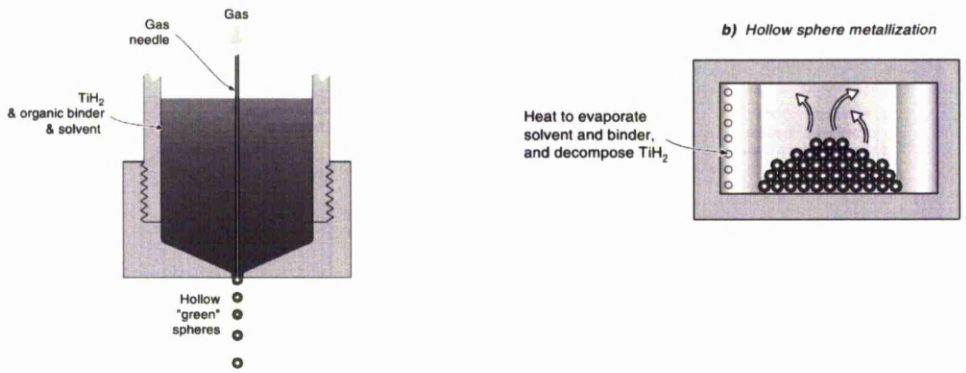


Figure 21: Manufacture of hollow spheres by blowing of disolved titanium hydride and binder. [21]

Figure 21 details an alternative method to Figure 20 to make a cellular metallic structure based on hollow spheres. A slurry containing a dissolved metal hydride and binder mix is blown through a nozzle creating hollow spheres that dry during a fall in a drop tower. Once collected the spheres are heat treated to dry off the remainder of the solvent, and decompose the metal hydride leaving the metal sphere based lattice with a porosity of as high as 95%. [21]

3.1.2 Metallic Open Cell Lattice Structures

The following described methods are used to produce open cell materials.

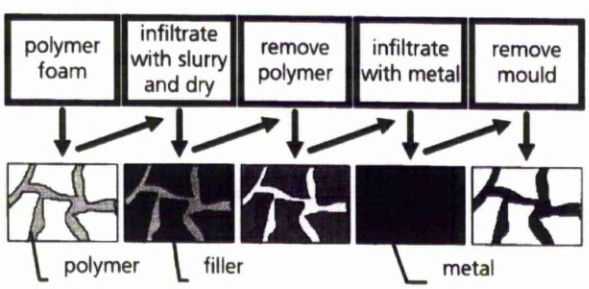


Figure 22: Cellular material by investment casting.[4]

Figure 22 shows a way of manufacturing lattice structures using a traditional casting technique, investment casting. This is where a model of the desired

component is encased within ceramic, and the model then melted, or burnt out of the ceramic. This forms a mould for the component. The ceramic mould is supported and the molten metal is poured into the mould. Once set and cooled the ceramic mould is removed leaving the remaining part. The master model has to be open cell as closed cells would result in solid ceramic inclusions in the lattice structure. Porosity ranges from 80-97% with cell sizes of 5–0.6 millimetres. This material is made by ERC in the USA with the trade name Duocel. The foam is comparatively expensive and the material properties of the trusses in the lattice are as cast, and so can be brittle. The company produces 8 cubic metres a year, and it has been made in aluminium alloys, copper, and magnesium. [4]

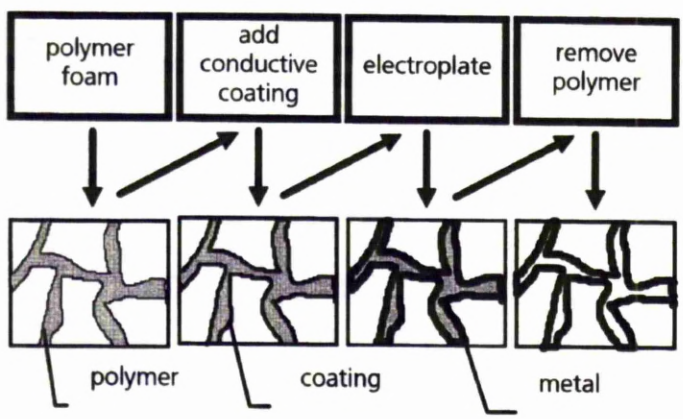


Figure 23: Manufacturing method of cellular material using electro-deposited metal onto a polymer foam [4].

Another two manufacturing methods making use of polymer foams are electro and vapour deposition. If electro-deposition is used the polymer foam needs to be prepared with a coating of a conductive material, for instance by dipping it in a carbon black slurry. The foam is coated with metal by either electro-deposition, or vapour deposition. The latter being performed in a vacuum where the polymer foam is the surface onto which metal vapour condenses. Figure 23 Details the

manufacturing process for creating an electroplated lattice, and Figure 24 shows a result of the process. Lattice structures with a porosity of up to 97% with cell sizes of 3.2 to 0.5 millimetres can be made using this process.

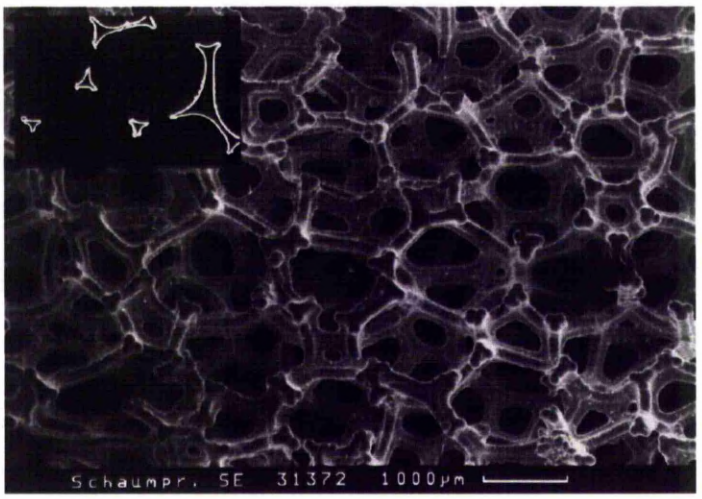


Figure 24: Nickel foam prepared by electro deposition. Inset showing thickness of metal on trusses. [4]

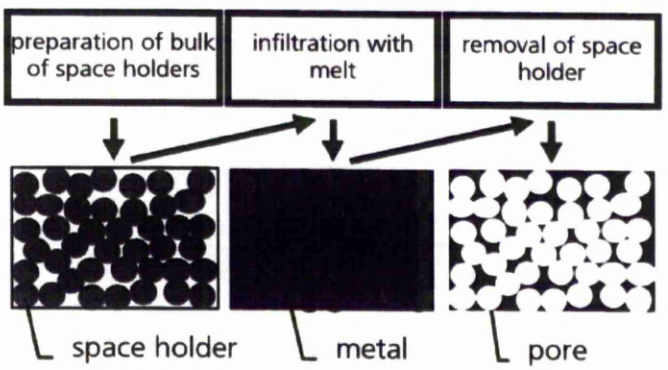


Figure 25: Cellular material using space holding fillers. [4]

Figure 25 shows a comparable technique to investment casting. The lattice is created in a mould which is removed though burning or melting out. The critical difference being that there is no master model. The mould is created using loose media which packs into a contained volume not filling it completely. Once the

loose space holders are packed in the melt is poured in and left to freeze. The loose space fillers touch each other, which results in a network of interconnected pores in the resulting cellular material. The porosity of these materials are limited to about 80%, but have very uniform pore sizes as this is dictated by the space holders [4].

3.1.3 Metallic Designed Topology Open Cell Truss Structures

Metallic foams provide a method for making cellular materials in large block sizes and are suitable for manufacturing on a large scale. However the materials produced are generally closed cell and have a large range of pore sizes, shapes, and in some cases occasional solid inclusions. Designed open cell lattice structures typically have porosities of 90% or over, and are far more predicable as the don't suffer the random distribution of pore sizes.

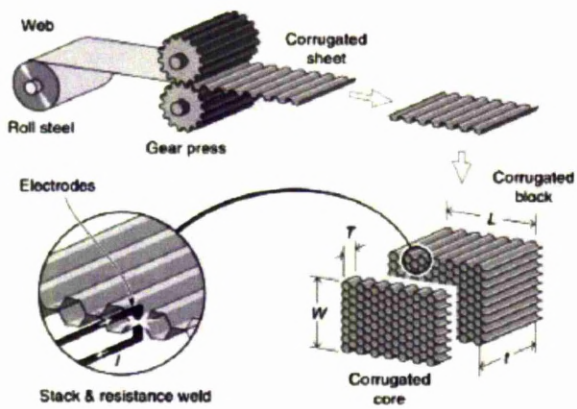


Figure 26: Manufacturing method for honeycomb structures. [24]

Designed open cell topology structures are mechanically far superior due to the truss elements in the structure can be arranged to avoid being loaded in bending. Figure 26 shows the benchmark honeycomb structure which is assembled from

sheet with a corrugation and welding step.

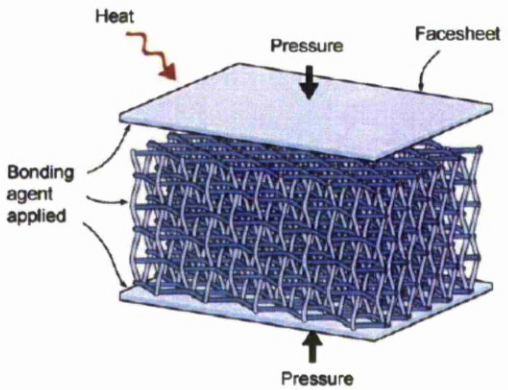


Figure 27: Metal textile laminate sandwich core [10].

Metal texile truss structures have been made for many years. To make these structures the raw metallic material is drawn into wires, woven into a three dimensional structure, and subsequently brazed forming nodes at each point where the wires contact. Figure 27 Shows the manufacture of a woven material based lattice truss core sandwich structure. The face sheets strengthen the core material in bending and in loading as the face sheet constrains the motion of the the outer trusses and also spreads the load across a larger area of the structure [9].

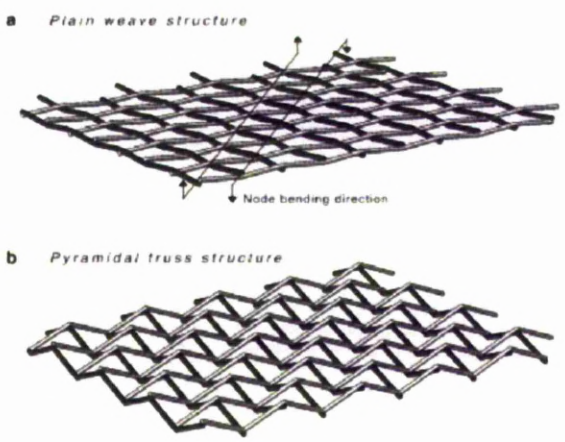


Figure 28: Method of creating a the topology for a metal textile core using wire mesh. [24]

Figure 28 shows the forming step of the manufacturing process for woven truss structures. This step forms the stock wire mesh into the required topology. The porosity of the structure is controlled by the relation of the cell size to the wire diameter. The ability of the wire to hold its formed shape suffers as the wire diameter to cell size ratio reduces. A less stable mesh is more difficult to control during manufacture assembly as the nodes have to match up with nodes on other formed meshes. In addition to this the thinner wires are less likely to hold their shape [24].

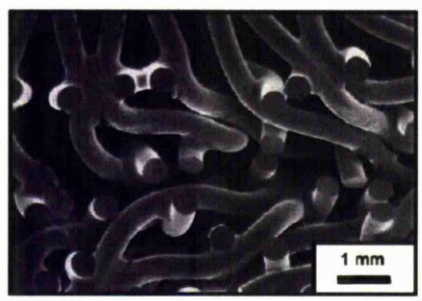


Figure 29: Deformed metal textile lattice core showing that energy is absorbed by the plastic deformation of the wire structure [9].

Figure 29 shows the joins made by brazing the woven structure. The structure shown has been deformed by an impact. The large voids and design of the weave results in a structure that can absorb a significant amount of energy when deformed. As the wire is drawn it has high yield strength. The design of the weave takes advantage of this by allowing the trusses to buckle and stretch absorbing energy in the process of its collapse [9].

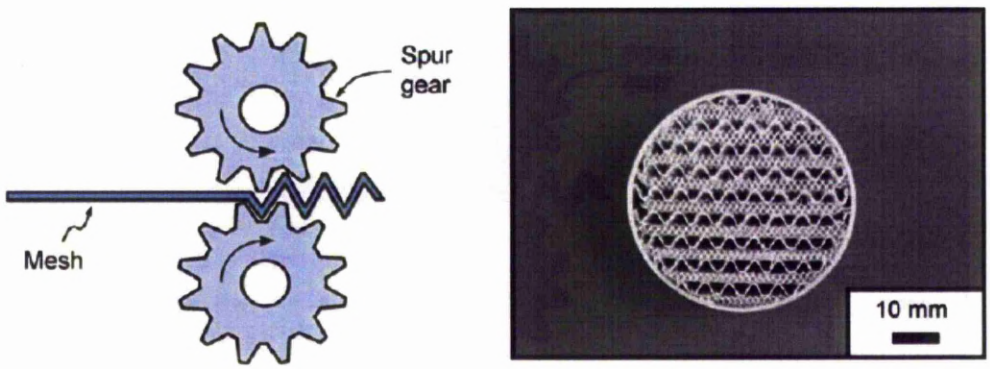


Figure 30: Forming method and example part of an assembled corrugated core [9].

One approach to get above 90% porosity is shown in Figure 30. It is challenging to get above this porosity value using woven structures. This shows layers of the metallic woven structure spaced out by layers of mesh that have been corrugated [9].

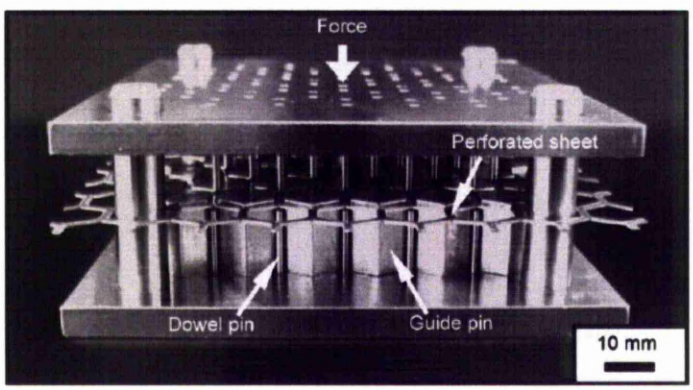


Figure 31: Part of the assembly sequence used to manufacture tetrahedral truss cores [9].

Figure 31 Shows part of a manufacturing method which uses sheet metal with stamped holes which is subsequently formed from a 2D structure into a 3D structure of truss links. Layers of these structures are subsequently aligned and brazed together with the possible addition of face sheets [9].

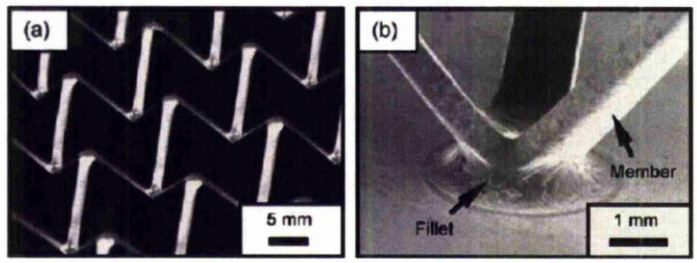


Figure 32: Pressed tetrahedral truss core and bond to face plate [9].

The cut and formed sheet is shown in Figure 32 along with the bond it makes with the face sheet when brazed. This structure provides the lowest relative density of solid material to void of the current methods for producing lattice structures. This is due to it using larger cell sizes than those seen in the woven materials. This is a possible alternative to honeycomb two and a half dimension materials.

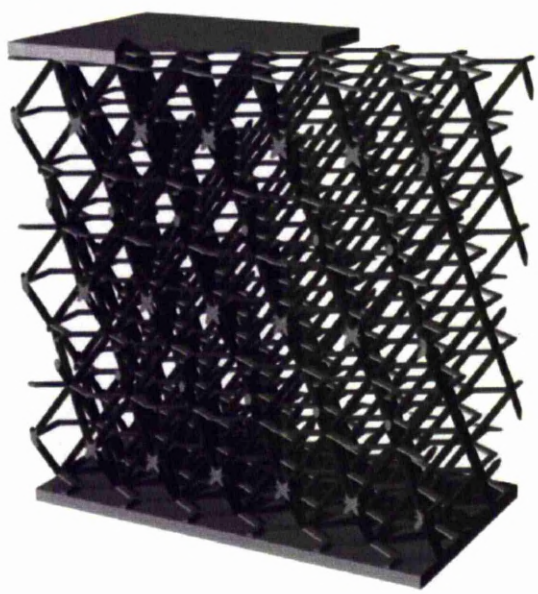


Figure 33: Computer visualisation of assembled truss structure with face sheets. [24]

Figure 33 shows a computer visualisation of an application of the deformed sheet lattice structure. The example shows many layers of the structure being used to form a core for a sandwich structure. Figure 34 shows one level of the lattice core

visualised in Figure 33. It comprises of two flat sheets with hexagon holes separated by a sheet formed in a method similar to that shown in Figure 31. The structure is then bonded together at the nodes.

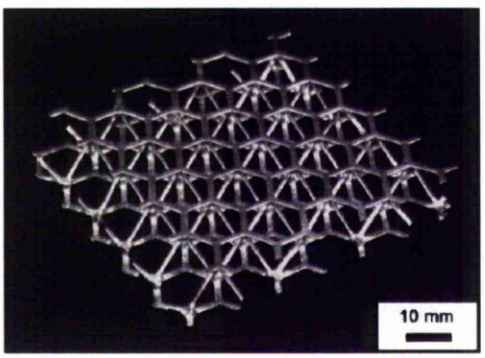


Figure 34: Completed tetrahedral truss core without face plates [9]

Similar structures can be created by investment casting which can form a wide range of cell topologies. The cores can be injection moulded, assembled, or built using layer manufacturing technologies. Either way the minimum truss diameters and cell lengths are restricted by the formation of casting defects on more elaborate cell geometries.

A similar technique was investigated by Brittain et al where the initial mesh is formed using electro deposition but the author suggests that the techniques highlighted above is more suited for larger cell sizes and suggests their method is more suited to the cell sizes in the order of a few millimetres [25].

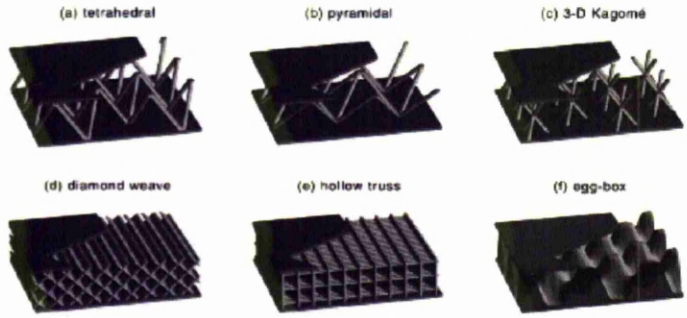


Figure 35: Computer images of possible lattice geometries created by investment casting. [24]

Figure 35 shows the application of prismatic two-and-a-half dimension open cell lattices and repeating topology structures to creating sandwich structures. The use of investment casting to create these structures is proposed and an example of a lattice sandwich structure created using investment casting is shown in Figure 36.

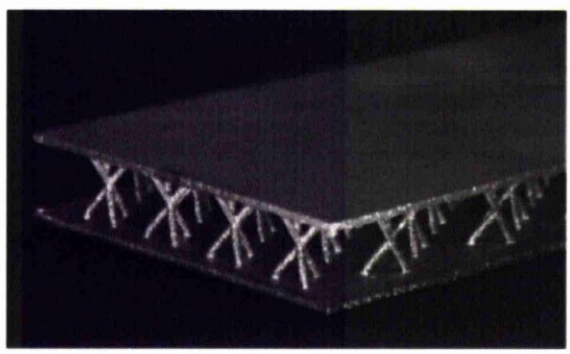


Figure 36: Designed lattice core created by investment casting. [24]

3.2 Selective Laser Melting (SLM)

SLM is used to create parts in a short time frame from design to manufacture, or components that would have been impossible to manufacture with traditional manufacturing techniques. It has been described in detail by many previous authors. A wide range of pure and alloyed metals are most commonly used, but some work has been carried out into SLM of ceramics [26]. The SLM equipment and process is mechanically very similar to Selective Laser Sintering (SLS) which is a powder based Rapid Prototyping (RP) technology commercialised in the early nineties.

This generic term RP is a little misleading as both machines are capable of manufacturing low batch size, or highly customised components, in many cases it is more suitable and descriptive to refer to SLM and SLS as machines based on Layer Manufacturing (LM) technology.

LM describes processes that typically create components by building them up in sequential layers. For clarity the direction in which the layers are stacked is generally referred to as the z axis, and the layers are aligned parallel to the x and y axis. When referencing the x axis this usually implies the one parallel to the observer when they are in front of the system looking in.

3.2.1 The SLM Process

SLM utilises a fibre laser to provide a source of focused infra red energy which is used to selectively drive a phase change in the powder in a pre-placed powder bed. In SLM the phase change in the powder causes the particles to fully melt and form a melt pool, whereas in SLS the powder particles partially melt, and coalesce with surrounding powder particles leaving a porous structure. It is worthwhile noting that even if a melt pool is formed there is no guarantee of forming a component of high density, as an understanding of the effect of varying the processing parameters is required to obtain parameters that fall within a processing window that creates components with high density.

In order for a part to be produced by SLM the raw CAD data must first be sliced at defined z levels to give a set of contours. The contours define the bounds of the solid region for the sliced geometry, and are often used as an outline scan to improve the surface finish of the components. In SLM the contours are processed further to create a scan path to fill the interior solid regions of the part. There are a number of options for the pattern used for the fill scan. During a build the SLM machine processes each layer and the scanners will direct the focus of the laser over the fill and contour scan paths, causing all of the powder within the contour to be melted which coalesces and subsequently freezes to form a solid region on the powder bed.

3.2.2 Current SLM Geometric Limitations

Parts built by SLM have far less limitations to their geometric shapes than those created by employing traditional manufacturing techniques such as machining, casting, and moulding. For example the parts can include re-entrant shapes to a

degree that would be beyond the capabilities of Computer Numerically Controlled machining centres (CNC). This is because the laser always has access to the full area of the part on each layer, and so can add to it on any location. There are however limitations to what can be built. All sections that are processed by the laser must either be above material processed in the previous layer, supporting structure, or the substrate itself. The thermal induced expansion and subsequent shrinkage is one of the factors that prevents building on a free powder bed. Curl has been observed between layers, and between scan tracks on the same layer. The forces involved can be high enough to cause cracks visible by low magnification optical microscopy. Research has been carried out by previous authors into building unsupported structures [27].

What can and can not be successfully built using an SLM machine is dependent on the processing parameters, part geometry, scanning strategy, and the design of the supporting structure. Software is available that will create support structures automatically, but it is rare that these will not require further modification by the user of the machine. If the settings are optimised solid geometries can be made with good surface finish at angles as low as 40 - 45 degrees to the x-y plane. Angles lower than this, or significant spans at this angle normally require additional support. Thin walls and other features have been built as low as 250 microns in thickness, and at the other end of the scale parts with significant solid sections such as press tool inserts in rapid tooling have been successfully fabricated in SLM.

The SLM process is currently not capable of building parts unsupported. This is due to the high melt temperatures of the working material and the processing

parameters that have to be adopted to create high density components. In polymer SLS (Selective Laser Sintering) to reduce the curling effect that results from processing consecutive layers the whole power bed is heated to within a few degrees of the melt temperature of the polymer being processed. This reduces the rate at which the layers cool immediately after processing, and maintains a higher temperature until the build is complete. To achieve this in SLM would require significant increases in the cost of the equipment to retain the current ability to process metals such as steel and tungsten. It would be very challenging to engineer a machine that could operate at a wide range of temperatures to suit metals from aluminium, right through to tungsten. Some authors have seen benefits of heating the substrate on which the parts are built. Other than the heat from the substrate (which is typically no higher then 300 degrees centigrade) the powder is processed at room temperatures. The molten polymer processed in the SLS process has a greater viscosity than the molten metal. The lower viscosity of the metal material means it is far more susceptible to the effects of Marangoni convections and metal melt pools also exhibit large surface tensions. Rayleigh observed the tendency of tracks of water to ball up in the late 19th century. His observation was that the tracks of water broke up into balls once the length of the track exceeded a multiple of its width. This has been demonstrated to have relevance in the SLM of metals by previous authors. [27] [28]

3.2.3 Rayleigh Instability

Rayleigh Instability describes the tendency for a cylinder of liquid of low viscosity to break up and form spheres in order to lower its surface energy. Figure 37 shows an illustration from Powers et al [29] showing how the break up of a cylinder propagates through the length of a liquid cylinder. Powers et al propose a front velocity where the liquid cylinder is formed and the break up follows the formation of the track. The first step of which is the formation of pinches in the track which subsequently pinch through the track breaking it up. In a later publication [30] Powers et al described a front of pearling that moved at a constant velocity, and one that is separate to velocity of the formation of the cylinder.

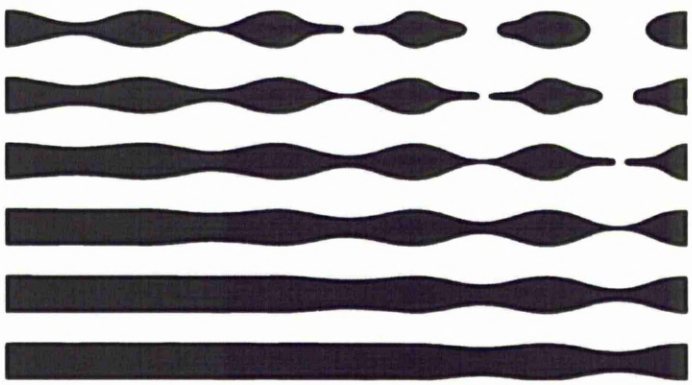


Figure 37: Time stepped illustration from bottom to top of the propagation of Rayleigh Instabilities. [29]

Grinfield [31] investigates the effect of substrates on the Rayleigh instability. The work previously discussed in this survey investigated the instabilities of a fluid cylinder that was in free-fall, and not in contact with a surface or substrate. Grinfield mentions that if the height of the cylinder (as opposed to it's length) is smaller than the radius of the cylinder then there is no bifurcation of the track. This is of critical importance to the understanding of the SLM process as when the

weld track is wetted on to the substrate, or previously melted powder (which has subsequently solidified) then the track will be less likely to break up than one on a free powder bed as the wetting to the substrate stabilises the liquid.

3.2.4 Marangoni Convection

Marangoni convection describes the motion of molten material driven by surface tension gradients, flowing from regions with low surface tension to those with a high surface tension [28]. The differences in surface tension are caused by thermal gradients in the weld pool. The addition of active surface elements such as oxygen or sulphur to the melt can cause the direction of the Marangoni Convection to reverse resulting in a change in weld pool shape. Amberg et al [32] discuss the effect the Marangoni flow can have on the shape of the weld pool and this is illustrated in Figure 38.

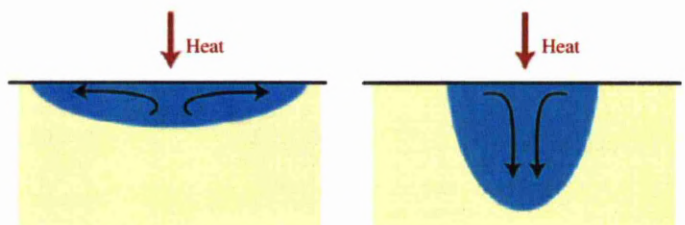


Figure 38: Posative and negative Marangoni flow and the effect on weld shape. [32]

Both Childs et al [27], and Morgan et al [33] discuss the Marangoni effect. Morgan et al discusses the use of the pulsed NdYAG laser to overcome it's effect and Childs et al discuss using it in the process to flatten the weld, but states that this approach would require fine control over the processing parameters leading to a reduced window of processing parameters which create the desired quality of manufactured SLM component.

Various authors have investigated the possibilities of using the Marangoni Convection to improve the qualities of the weld such as Shanping et al [34], Fuji et al [35], and Amberg et al [32]. Shanping et al investigated adding carbon dioxide into the argon shielding gas and noted that this increased the oxygen content in the weld metal. The authors noted two thresholds which marked distinct changes in the shape of the weld pool. The first was from a wide and shallow pool to a narrower and deep pool when the carbon dioxide content in the shielding gas exceeded 0.2%, and the oxygen content in the weld exceeded 100ppm. The second threshold occurs when the carbon dioxide content exceeded 0.6% and the oxygen content of the weld reached 200ppm. At this upper threshold oxides form on the surface which produces a barrier preventing further oxygen entry into the weld pool. At this point the shape of the pool reverts back to a wide and shallow pool, but with the base concave, rather than flat as was the case when the oxygen content in the weld was below 100ppm. These shape changes are shown in Figure 39.

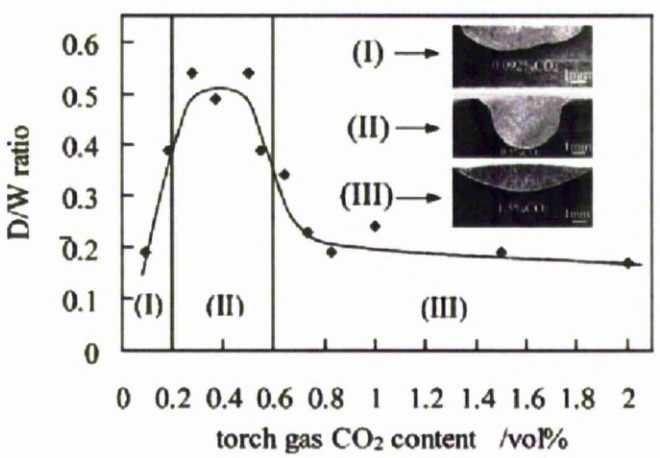


Figure 39: Depth to width ratio of weld with increasing carbon dioxide content in shielding gas. [34]

These distinct shape changes were also reported by Fuji et al as shown in Figure 40, where the authors investigated the addition of carbon dioxide to a helium shielding gas. In addition to this the author also investigated replacing the carbon dioxide content with oxygen. The differences between the authors reports where that Fuji et al reported a lower critical oxygen content in the metal that created a inward Marangoni convection which was 70ppm. More critically Fuji et al and Shanping et al give conflicting explanations of the reasons for a return to the wide weld pool when the second threshold is crossed. Fuji explains this as the Marangoni Convection returning to an outward direction. Shanping explained the change in shape after the second threshold was due to a thick oxide layer on the weld that creates a liquid/oxide interface rather than a liquid/gas interface which would suggest the Marangoni Convection is not the main factor to consider in shaping the melt pool in these regions. Shanping et al said that in inward convection still exists in the centre of the weld where there is a liquid/gas interface.

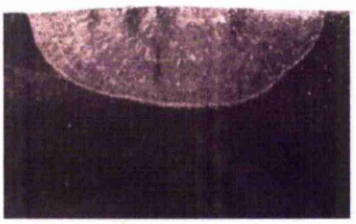
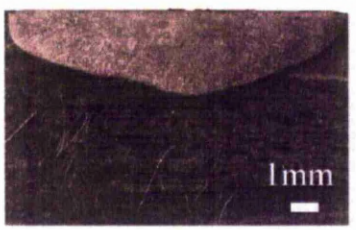
Shielding gas	Weld shape
Pure He	
He-0.4%CO_2	
He-4.0%CO_2	

Figure 40: Weld shape with increasing percentage of carbon dioxide in shielding gas [35]

Shanping et al did show some data on the effect of the oxygen content of liquid iron on the molten irons surface tension. It can be seen that the oxygen content has a far more significant effect on the weld shape than the temperature in the 250K range from 1873K as shown in Figure 41.

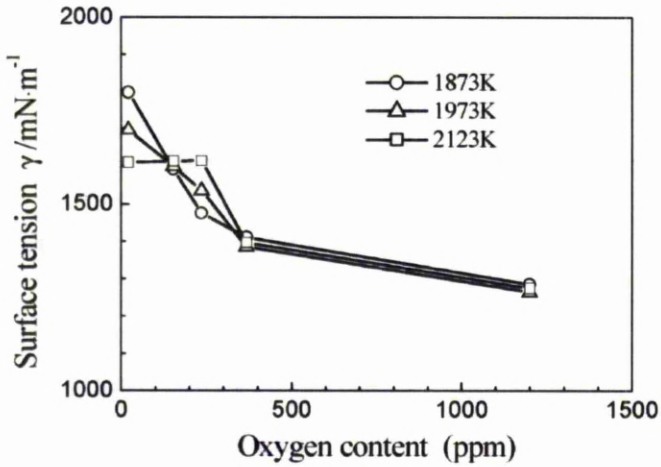


Figure 41: Effect of oxygen content of liquid iron. [34]

Amberg et al [32] also drew attention to the effects of the distribution of the surfactants on the weld pool which have an effect on the surface tension, giving sulphur as an example. Amberg et al also discussed the Marangoni Convection re-distributing the surfactants which in turn causes a reversal of the convection causing the surfactants to be redistributed only for the cycle to repeat. Unfortunately there is no indication as to the time scale for these reversals. This sequence is detailed in Figure 42.

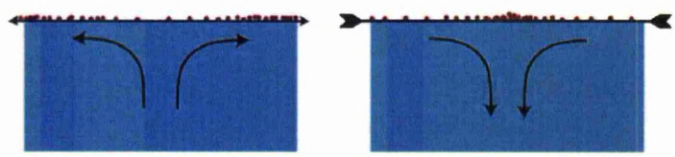


Figure 42: The effect of surface elasticity on Marangoni flow.[32]

3.2.5 Pre-placed powder bed & thin wall scan tests.

To gain a better understanding of the SLM process previous authors have used single scan lines on a free powder bed, hatching on a free powder bed, and consecutive scan lines onto the substrate. The experiments with single scan lines [27] showed good correlation with models used to predict when the tracks would break up because of the tendency to ball up due to the Rayleigh instability. In addition to this the papers also identified some common patterns. Childs et al 2005 [27] identified six different results from scanning on a free powder bed; nothing at all, continuous and flat/concave topped, continuous with rounded tops and slightly sunken into the bed, occasionally broken, frequently broken (balled), and fragile sintered tracks. These results where mapped onto graphs to give processing windows to show the laser power and scan speed used to process the links. These graphs showed similarities to Morgan et al 2001 [36]. Their work was mainly investigating the effect of a pulsed laser system and looking at processing square sections on a single layer using bidirectional raster scanning, whereas Childs et al 2005 [28] looked into the raster scanning of square sections on a free powder bed.

Morgan et al did seem to disagree with Childs et al in stating that "Typically, the viscosity of a metal is an order of magnitude higher than that of a polymer. This significantly reduces the tendency of the metal to flow in it's liquid state, causing regions of porosity across the layer" [36]. Childs et al's results showed a significant tendency of the melt to move, this was shown by the creation of small pools of melt in front of the melt pool that periodically flipped from in front of the beam/powder bed interaction point to behind and into the main melt pool, and the

reduction in length between the vectors scanned and the resulting metal track. The reduction in length was beyond what could be explained by contraction alone.

Niu et al 1999 [37] investigated differences in single scan lines behaviour on free powder between gas atomised and water atomised powder. This paper showed that increased oxide levels increased the balling of the track. This could be explained by the Marangoni Convection discussed in a previous section. The effect the oxides have on the characteristics of the melt pool, and subsequent mechanical properties of the processed parts (as well as a tendency for some metallic powders to combust when processed under atmospheric conditions) necessitates the use of a shielding gas to purge the working environment of the SLM equipment. The previously discussed work on the effects of Marangoni Convection from Shanping et al [34] and Fujii et al [35] and Childs et al's [27] expression for a preference for flat scan tracks in SLM would suggest that the oxygen level in the process chamber should be minimised as far as possible in order to avoid the oxygen content of the melt exceeding 70-100ppm.

Childs et al [27] presented stills from a video sequence as shown in Figure 43 demonstrating the formation of a secondary melt pool in front of the direction the laser focus was being moving, and theorised that this was caused by reflections from the leading edge of the melt pool. The captured video images which where presented and the description of the phenomena suggest a periodic pattern. As the melt pool establishes a uniform volume after the start of the scan vector it may interact with the laser beam where it intersects with the surface of the powder bed. This would cause some of the energy that would have been absorbed into the powder bed to be reflected, hitting the powder in front of the laser/powder bed

intersection. As this situation continues the additional laser energy forms a secondary melt pool as shown in the images. The secondary melt periodically coalesces with the primary melt.

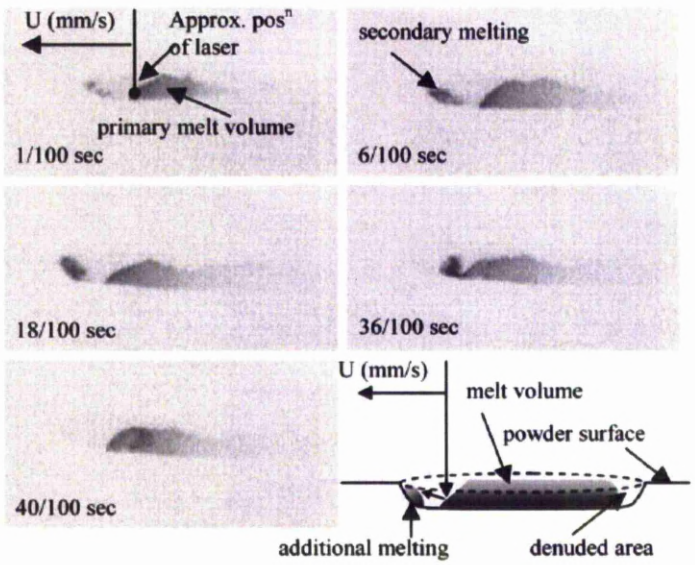


Figure 43: Video stills from Childs et al detailing the formation of a secondary melt. [27]

A few authors have noted periodic [28], [33] width changes in thin walls and patterned appearance of the uppermost layers of square test parts that have been scanned on free powder beds. This periodic variation in scan track size is also seen in the micro-structure of the thin wall as shown in Figure 44. The periodic pattern of creation and absorption of a secondary melt into the primary melt may offer one explanation for this pattern. The pearling pattern due to the Raleigh instability discussed by Powers et al [30], [29] may also explain this, but with the weld freezing before it could completely break up into separate weld pools. This however would not be supported by Romobouts et al [38] who suggest the time period required for the breaks to form is in the order of 0.5ms, far less time than it would take the weld to freeze. Morgan et al [33] observed that if the scan length

was less than the wavelength of the periodic width variance in a thin wall structure the walls where smooth with no evidence of the pattern.

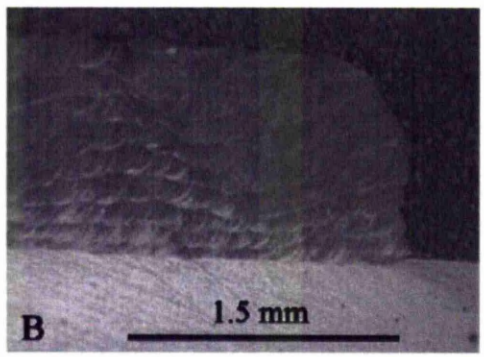


Figure 44: Periodic pattern in the microstructure of thin walls.[33]

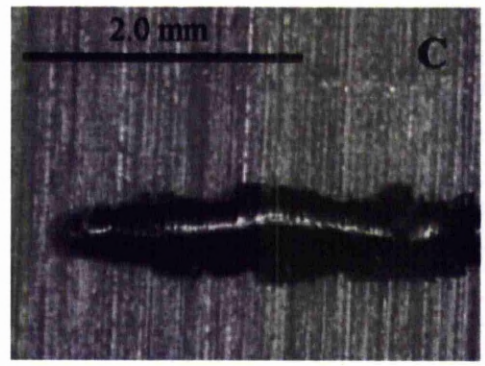


Figure 45: Thin wall showing a wall thickness of approximately 0.5mm. [33]

Looking at Morgan et al's images of a 2mm scan track in Figure 45 the scan width appears to be about 0.5mm. Due to the Rayleigh instability this would tend to break up if the melt extends beyond 1.57mm. This would suggest that this wave phenomena may be driven by the early stages of the pearling and pinching described by Powers et al [[30], [29]]. As the substrate to which the weld (or liquid cylinder) is wetting has been shown in models to have an influence on the break up of the cylinder one would assume that once a periodic variation has been established in a thin wall or solid it would propagate though its height, slightly offset on each layer as may be the case with the patterns on the side of the thin walls. One other periodic pattern has been examined in models by Amberg et al [32] is the flow within the melt pool reversing due to the build up of surfactants, but there was no mention on the time scale for the elastic motions.

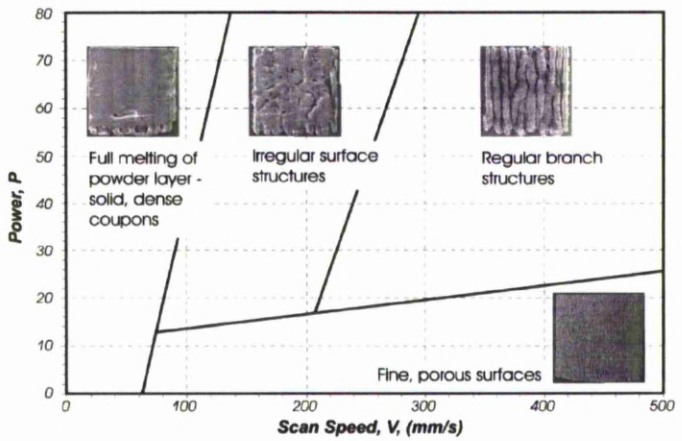


Figure 46: Process window as indicated by Morgan et al. [7] Laser beam size was 80 microns and powder range was 0.5 to 56 microns.

Further comparisons between Childs et al's [27] and Morgan et al's [36] show some similarities for their process maps Figure 46 and Figure 48. Child et al split the results for the single scan lines into six groups as shown in Figure 47. Track types B through E seem to occupy the same locations as the four results shown on Morgan et al's process parameter window in Figure 46. The squares shown are bi-directionally raster scanned from left to right. Scan type A - the flat continuous track – and scan type F – no effect on the powder bed – are not present on the Morgan et al's process map. Scan type B, the continuous dome topped and sunk track could be compared to the "Full melting". Scan type C could be compared to the irregular surface structures token. Although the track appears uniformly broken up in Figure 47 the author comments that this is not always the case. Scan type D corresponds to the "Regular branch" structure as the break up is at a much shorter wavelength as is seen in the token. Finally Scan type E corresponds to the "Fine porous" token. This appears to be the area on the process maps where by the powder is sintered, rather than melted which would result in a weak porous structure as described.

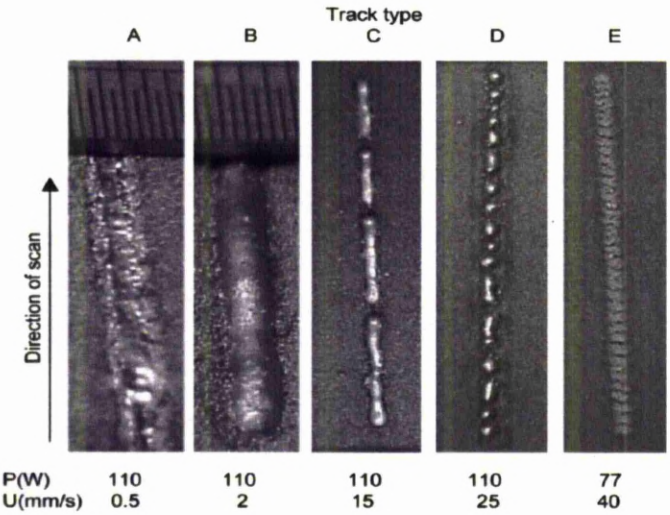


Figure 47: Five distinct results for scan lines on a free powder bed as presented by Childs et al. [27] The Sixth type, no result is ommitted.

What is interesting to note is that other than the powder used the conditions in the experiments are very different Morgan et al's work works with YAG laser power in the range 0 – 80W, a spot size of 0.08mm and scan speeds between 0 – 500mm/s whereas Childs et al was looking at 0 – 200W carbon dioxide laser, a 0.55mm laser spot size, and much slower scan speeds of 0 – 50mm/s.

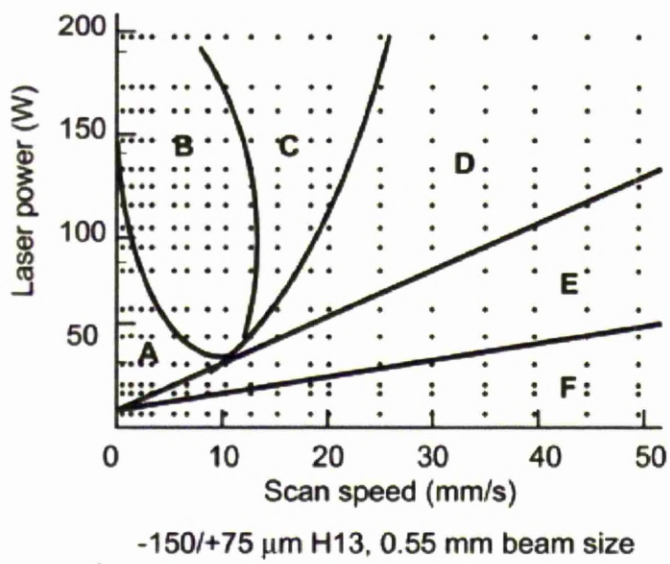


Figure 48: Process window as identified by Childs et al. [27]

Figure 49 shows the results of some raster scans on a free powder bed as discussed by Childs et al [46]. On the left this shows the periodic pattern seen on the "Regular Branch" square in Figure 46 and it corresponds to scan type D on Figure 36 further confirming the comparisons made earlier.

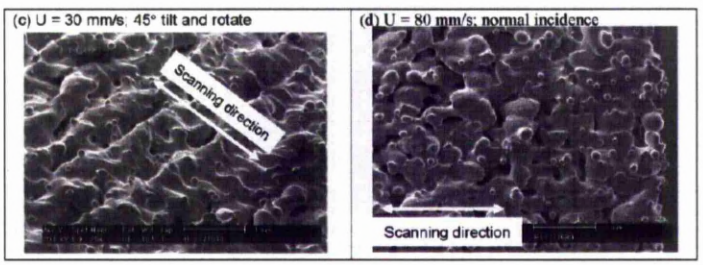


Figure 49: Two images from Childs et al showing results of raster scans on a free powder bed at 77W. [46]

On the right hand side of Figure 49 a result is shown that does not appear to be discussed in the paper. With bi-directional raster scanning the laser tracks from one side to the other, then reverses direction and scans slightly over lapping the previous scan track. When the track first doubles back the previous scan tracks temperature will be much higher than at the end of the return scan. This may explain the densification seen on the edge of the token shown. A reason for using bi-directional scanning is to increase the productivity of the scanners as they spend less time not scanning, where as if the scans where all in one direction the scanners would have to make a jump to the start of each scan line after finishing the last. It can also be used to reduce the problems associated with a leading edge pulse of laser power when the laser turns on, or to avoid a large settle time for the scanning system with the laser on. If the laser is switched off for the jump then there tends to be a high peak power at the start of the first pulse which can effect the powder bed. As the scanners make the jump there will be a settle time for the

mirrors before the system can continue processing the scan vector. If the laser where to remain on (to avoid the first pulse peak) then the laser energy would be focused on one region while the mirrors settle which could start processing the powder before the scanners have started the vector to be scanned.

3.3 Literature Survey Summary

Development of lattice structures is being lead against a number of distinct aims such as; achieving low relative densities, and designed or controlled lattice geometries. The current methods of creating designed structures can require many steps in the manufacture. For example one discussed method requires holes be punched out of sheet metal, it be shaped in a die, assembled on other stamped sheets, then brazed into one assembly. Casting techniques have also been employed to create structures but the cell size reported are typically in the 10mm scale.

Both of these techniques have severe limitations in themselves. Casting can require temperature profile control of the mould and melt, runners and risers, and the geometry itself needs to be designed to minimise the risk of trapped gas, and maintain a good flow through the mould. The stamping method has limitations on the angles of the struts, re-entrant shapes for example would mean the sheet metal (if it formed without tearing) would not be able to be removed from the stamp or dies – not to mention the stamp and die would collide.

While SLM has it's own limitations the research into these limitations is a far more recent and less mature field of research. At the time this research commenced work had not been carried out into verifying how the limitations seen

on more substantial structures effect the processing of lattice structures. For example it was not know if the lattice structures would be more or less tolerant in creation on the SLM machines.

The reviewed literature has shown significant previous research into the formation and movement of the melt pools during the SLM process. It was expected that better management of this melt pool would be key to addressing any geometric limitations of the structures.

An area of research that is notable in it's absence is detailed discussion of the software developments required to best create these structures. While research is published with some detail on the software techniques used to generate computer models of lattice structures, developing the software on and close to the machine to better create the structures is rarely discussed in detail. This may have been due to the commercial interests of the bodies carrying out the research. This research aims to clearly discuss the tools that are developed through research and validation of theory to enable following researchers to spend less time developing the tools and more time investing the potential of the created structures in terms of mechanical capabilities, variance in manufacture, and commercial viability.

Wang et al [39] did discuss the development of a parametric technique to create a CAD model conformal lattice structures and discussed the use of their software to create STL files of lattice structures with 20,000 elements. As will be discussed in the methodology section this technique was not workable using the provided experimental equipment as it wasn't able to process STL files at that level of complexity. Figure 50 shows a lattice structure created by Wang et al that contained 20% solid material in comparison to the solid CAD model that the

lattice structure filled.

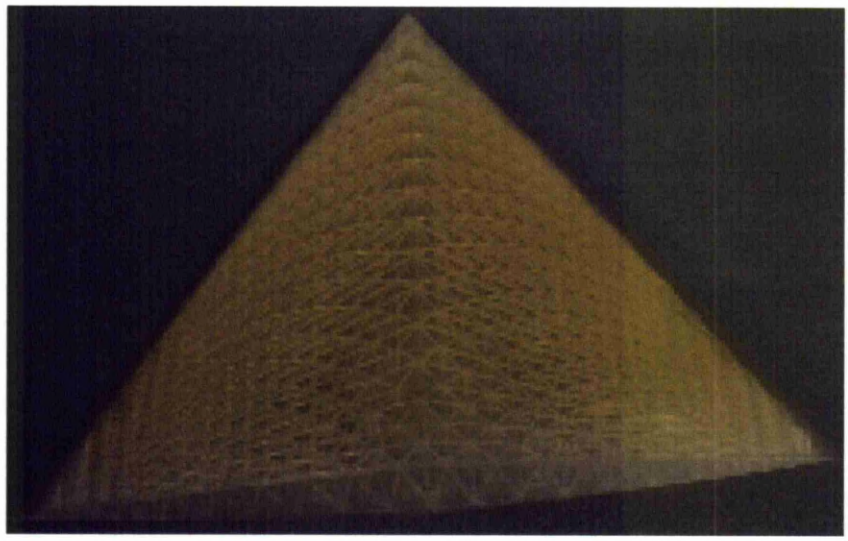


Figure 50: A lattice structure created on a Stereo Lithography machine using developed parametric modelling techniques. [39]

Wang et al and Williams et al [40] is referenced at a number of points through this thesis and while the research efforts are being led from different angles it is presumed that if continued the research in this thesis would develop the SLM system into a production method that is capable of handling the structures that they are developing. Wang and Williams are working from a view point of trying to create a specific lattice, or genre of lattice structures while this research focuses on developing the SLM technique towards realising it's maximum capabilities.

4 Developed Aims for Research

A wide range of applications for metallic cellular materials have been identified in the introduction and literature survey of this thesis. The range of applications include heat transfer, impact absorption, load bearing structures and implantable devices.

The current methods for creating metallic cellular materials range from open and closed cell foams with various degrees of random variations in pore sizes and solid inclusions through to lattice structures that have been designed and either cast or assembled from stacks of pressed sheets brazed together. What is very apparent is the applications for each manufacturing technique are limited by the utilised technique. For instance the strut dimensions in a lattice structure created from an investment cast technique are limited by the flow characteristics of the metal and the need for runners and risers.

SLM technology is in it's infancy and has only been commercially available since the mid nineties, and while it does suffer limitations they are potentially less limiting than other manufacturing techniques. A significant proportion of the limitations encountered prior to the research detailed in this thesis were due to perceived limitations such as the angle to which lattice structures could be built, the minimum size of elements, and the complexity of structure that could be built without requiring non-standard computer equipment.

The initial aims of the research that were set out and subsequently achieved and detailed in this thesis are thus:

- Development of a technique for creating the lattice structures that allows

complex structures to be built on the supplied experimental equipment. The developed tools are usable by collaborative researchers without significant training, knowledge of programming, or using the command prompt / terminal window. This will allow subsequent research to focus on the capabilities granted by this research rather than the technology.

- Investigation into the limits to which elements in the lattice structure can be built in terms of their angle to the horizontal. This knowledge was subsequently used to demonstrate the capabilities of the SLM process after modification of the software used to run the experimental equipment.
- Research the required methods to create horizontal links on one layer of the build. Aside from unsupported structures this was perceived as the most difficult aim. This was achieved as detailed later in the thesis where horizontal links where formed between two vertical pillars 5mm apart. Suggestions are made as to how the mechanical properties of these links could be improved upon to make the created links an attractive prospect for inclusion into materials made using a lattice structure with horizontal links.

This set of aims provided an initial direction for the research which increased in scope as the developed techniques were utilised by collaborative research projects.

5 Methodology and Experimentation

5.1 Experimental Equipment and Operation

5.1.1 SLM Hardware

The SLM equipment utilised for these experiments was an MCP RealiZer machine. The main components of which comprise of a 200W infra-red fibre laser manufactured by SPI, galvanometers from Cambridge Technologies, SIL Optics F-Theta lens, and two computers. One of the computers runs a windows XP operating system, is used as a user interface and allows build preparation - which can also be performed remotely. The other runs a DOS operating system, acts as a slave machine real time controlling the machine hardware, laser, and optics. The selection of this equipment rather than using suppliers such as EOS, and Phoenix was not part of the scope of this research as this was the machine that was available.

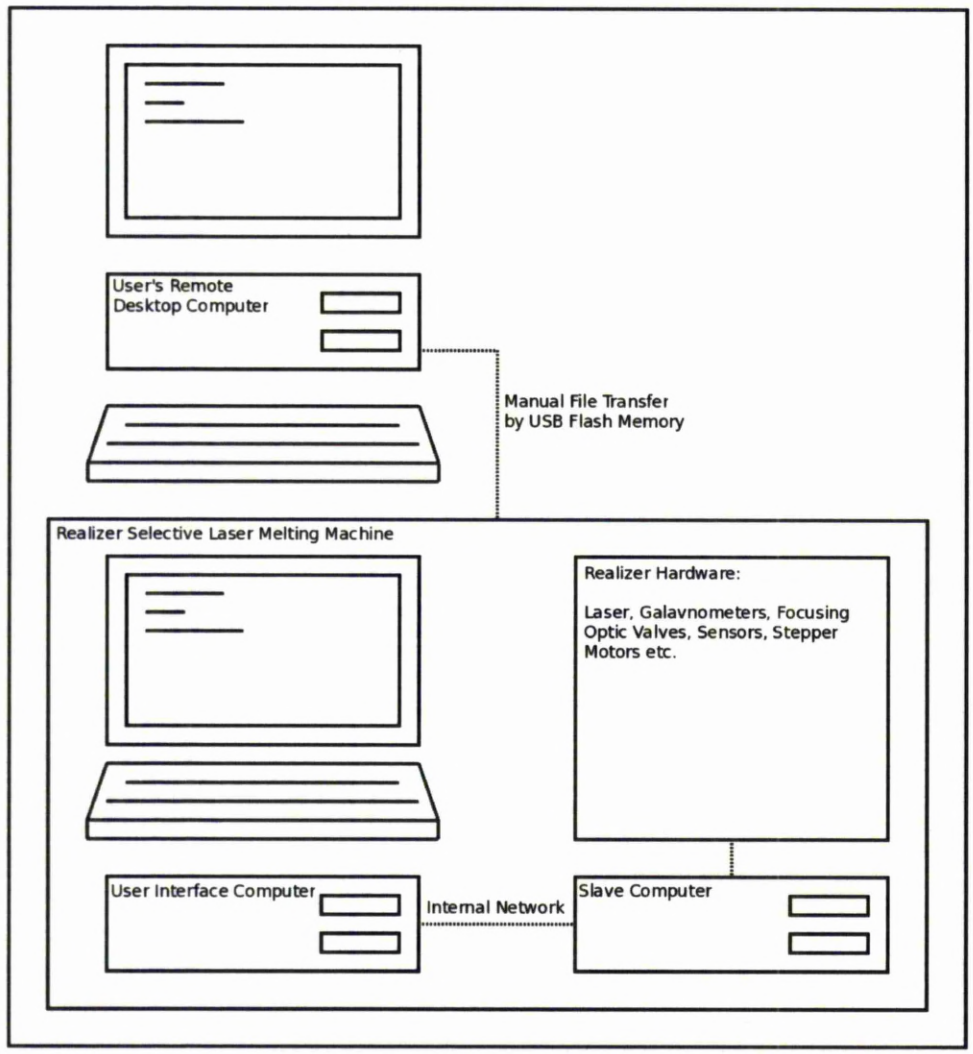


Figure 51: Computers used in building parts on the SLM equipment.

In the process chamber (excluding the gas circuit equipment) there is a powder cassette, elevator, f-theta optic, powder silo, front and rear overflow vessels, and the chamber door as detailed in Figure 52.

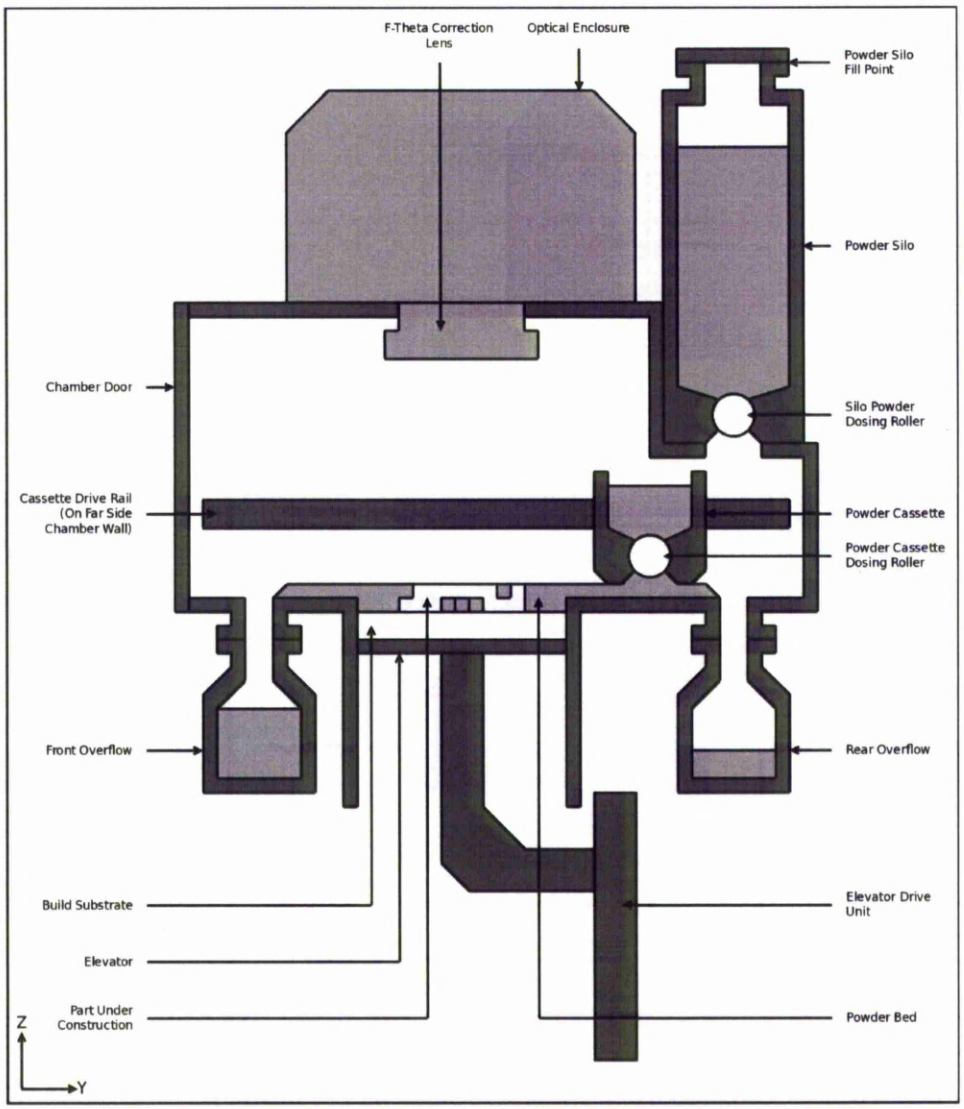


Figure 52: RealiZer mechanical layout.

The powder cassette comprises of a stepper motor belt driven slotted roller. The powder is metered onto the substrate by the slotted roller. The user can vary the amount of powder dropped by varying the number of steps the stepper motor driving the belt is driven on each add layer sequence. The cassette houses powder level sensors. On detecting low powder level the cassette is filled with powder metered out of the main silo. This takes place as part of the next add layer process. Any excess powder from an add layer, or cassette fill operation is collected in the

removable overflow vessels.

The gas circuit is used to maintain a low oxygen level in the process chamber and to remove suspended metal from the process gas. This is shown in Figure 53. The metal can become suspended during the add layer process, or from being ejected from the melt pool site mid process. The ejection from the melt site is either powder blown from around the melt pool, spatter ejected from the pool, or very fine particles from condensed metallic vapour.

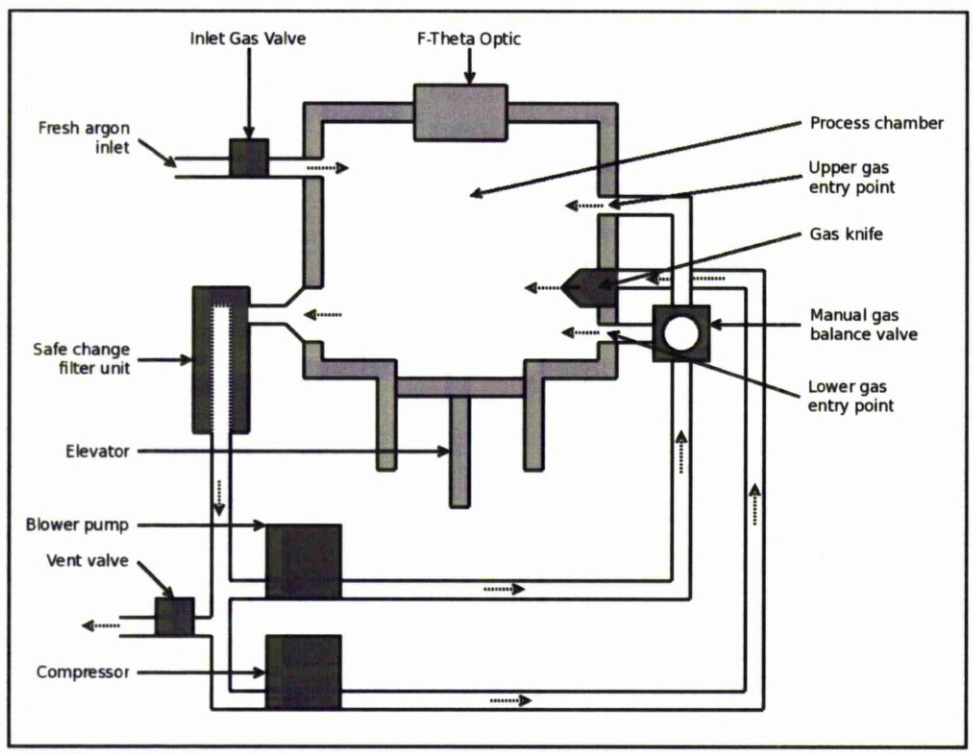


Figure 53: RealiZer gas circuit layout.

The initial chamber preparation is achieved by the user manually adjusting a flow control valve (not pictured) so that a slight positive pressure is maintained within the process chamber with the vent valve open. Once a desired oxygen level is achieved the vent valve is closed, and the recirculation pump activated. The argon flow is adjusted (the chamber leaks so always needs an argon flow to maintain

positive pressure) so that the same chamber pressure is maintained. This will cause the chamber oxygen level to raise slightly as the atmosphere inside the chamber mixes and all the oxygen that was trapped in the gas circuit is flushed out. Once the desired oxygen level is again reached and becomes stable the flow is dropped to a minimum value while still maintaining a positive pressure. Once the user is satisfied with the oxygen level within the chamber the build process can be started.

5.1.2 User Perspective of Machine

To create parts on the SLM equipment the user must first use the available software to generate the slice data in a format the machine can process. In order to achieve this 3D CAD models are converted into a surface mesh of triangles and sliced by the native SLM build preparation software, supplied by the machine manufacturer.

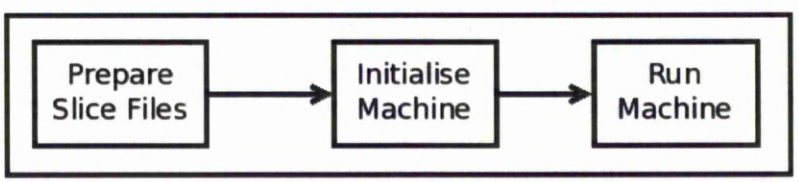


Figure 54: Flowchart detailing a simplified procedure fro running a build on the SLM equipment.

During the build preparation procedure the operator must ensure that all overhanging surfaces that are below a critical angle from the horizontal are supported, and processing parameters are assigned to the part which will result in a part with the desired density and visual appearance. This can be carried out by the user on software supplied by the vendor for use with the machine, or on a number of other software packages from alternative vendors. This process is

repeated for each part that is included in the build. Care must be taken to align the part in a suitable orientation so that the surfaces which require the best surface finish are upward facing, are not at a shallow angle to avoid excessive stair stepping, and that sudden significant section changes from one slice to the next are avoided where possible. The orientation of the parts has a major effect on both the build time of the process and the resulting part quality. Once the parts are orientated and supported they are copied to the RealiZer, assigned a material file, sliced, hatched, and stored.

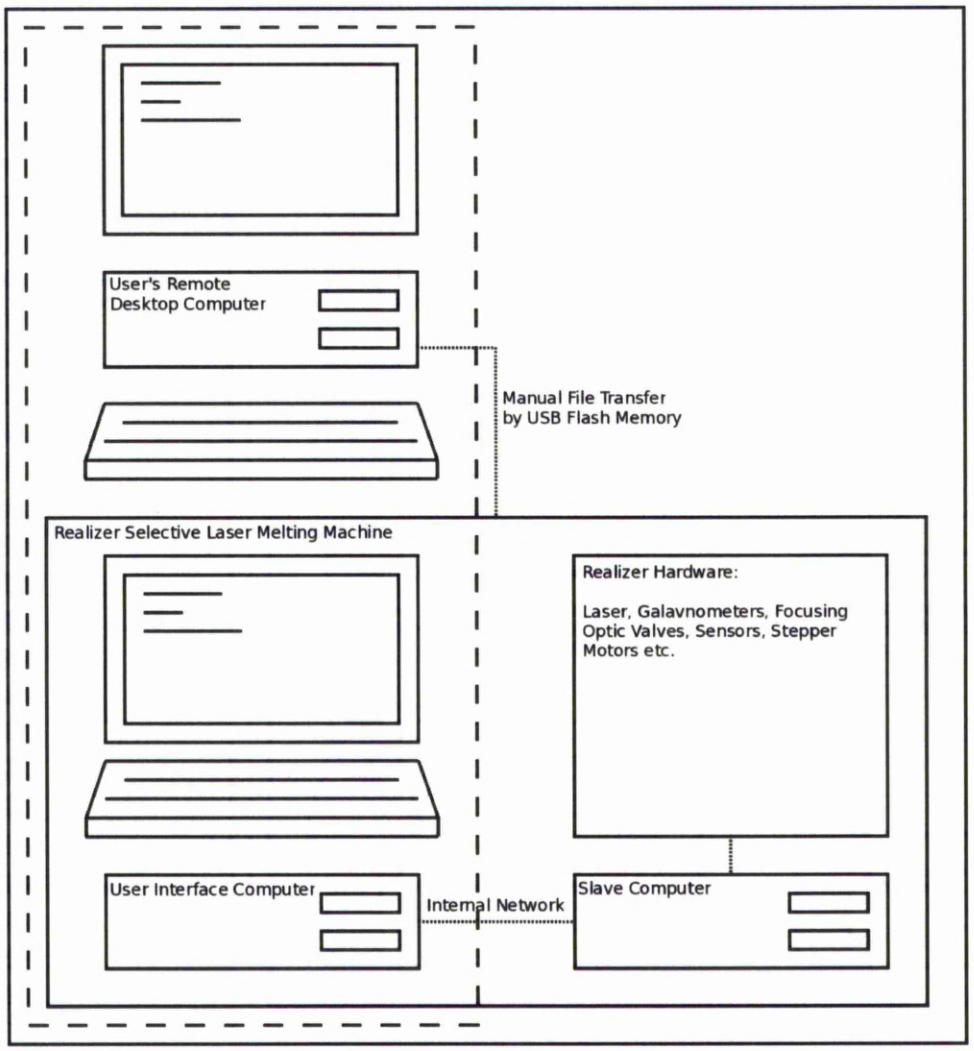


Figure 55: Computers used in building parts on the SLM equipment – identifying the computers which can be used for build file preparation.

In order for the machine to process the slice data the user must prepare a build. The build contains the slice data of all the parts, additional supporting, and labelling geometries to be fabricated. The information in a part's slice file is a collection of contours over a range of z-heights that have been derived from the intersection of a plane (normal to the z-axis at a given z-axis intersection) and the user provided closed-surface model (which means there are no holes in the triangular mesh) CAD file.

The sliced files are subsequently aligned in a build space within the RealiZer software and an assembly file is stored that lists slice files and the location that they are to be built in relation to the centre of the elevator. The build file at this stage is fully prepared and ready for use on the machine.

The machine is subsequently initialised prior to the build stage. During this subset of tasks the user follows through a set routine of maintenance tasks and set-up procedures prior to sealing the process chamber and purging the chamber of oxygen. The control is achieved through the RealiZer software that is also used for the slicing, hatching, and build preparation.

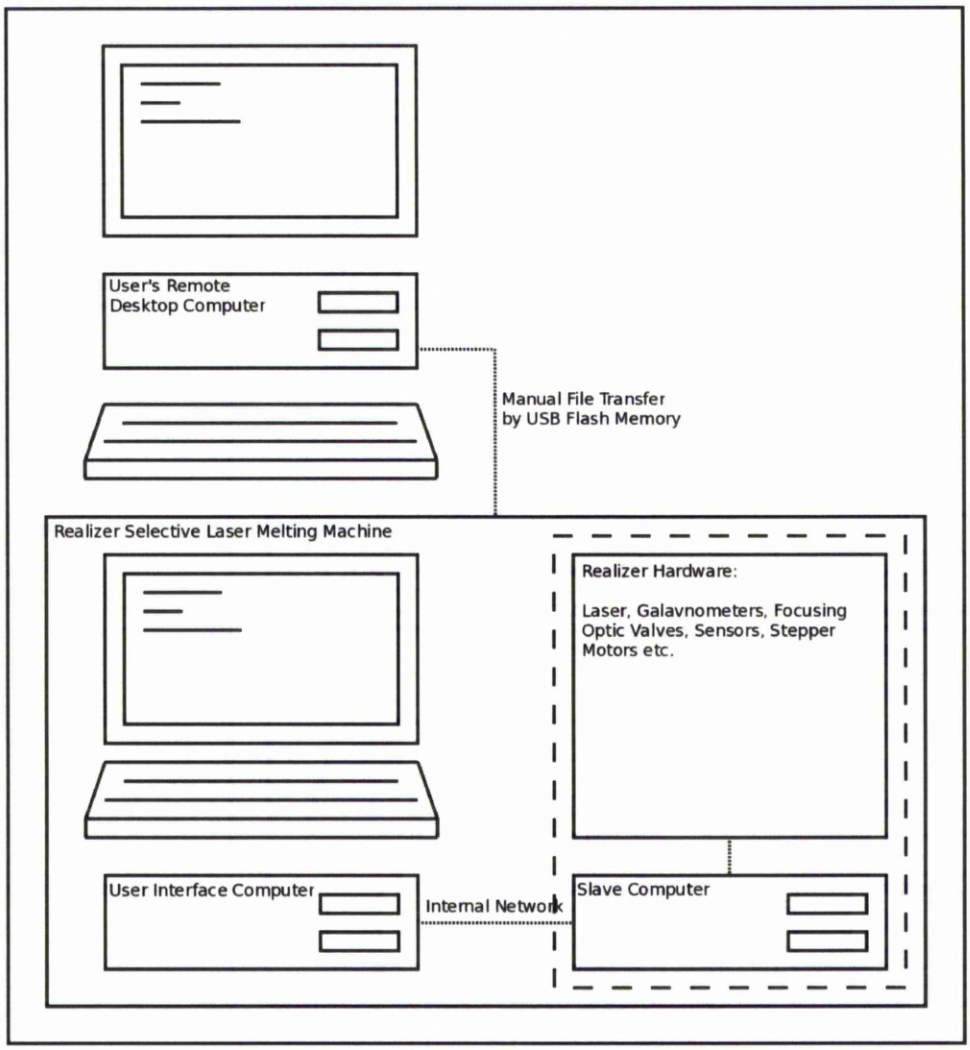


Figure 56: Computers used in building parts on the SLM equipment – identifying the computer used for real time hardware control and sensor data collection.

The build process consists of the machine sequentially processing all the slices of the supplied build. On each layer there may be slice data from a number of parts. The slice data on a layer, from a given part may consist of a number of contours, some of which could contain hatch data. Contours define the outer edge of an intersection with the part on that slice, and the hatch data is used to fill the inner region of the slice.

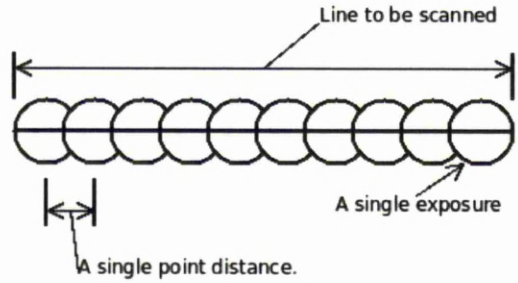


Figure 57: Detailing the method the SLM Realizer uses to scans vectors.

The SLM equipment traces a single hatch line, or single vector in a contour by breaking the line up into a number of points a distance apart (p-dist) defined by the user as detailed in Figure 7. The scanners aim the beam at the first point, enable the beam, remain at this point for a defined exposure time (exposure) then moves on to the next point for an exposure time, and repeats this until the scanners are aimed at the last point on the line. Following the exposure of the last point on the line the laser is disabled.

The scanning method gives four parameters that can be adjusted to control how a line is scanned on the part bed; the p-dist, exposure, laser power, and laser focus. There are a number of different hatch and contour types (collectively referred to as geometry types) that each refer to a particular set of these four parameters. This is necessary as different processing parameters are required for processing outer surfaces to hatching over the inner volume. Different parameters are required to get the best results when processing areas which are over volumes of powder that was not processed in the previous layer. Where open-surface style supporting structures are used these also require different parameters than closed-contours whose volumes are filled with hatch lines as support structures are formed with a single pass of the laser and will not be partially remelted in the same layer. This is in contrast to the hatch lines, or filled closed contours. It is usual practise to find

suitable parameters through experimentation then use gradual optimisation over many builds.

5.1.3 Machine Software

The SLM equipment is controlled by two computers which interface with a range of sensors, motor drivers, valves, and scanner drivers. The computer which controls the hardware and sensor interfaces is refered to as the slave computer. The other computer in the system is the user interface and is responsible for feeding the slave slice data. This data is used during the real time control of the laser and optics as the build process is running. During the initialisation stages of the build the user interface machine can be used to send 'atomic' commands to the slave, for example opening the gas vent valve. These atomic commands are commands that are passing a single instruction to the machine, rather than a command that is instructing the machine to perform a sequence of instructions. Once the slave has received the start build command and has requested the first slice the user interface computer only processes slice request, error, or stop messages and only responds with slice data, slice ready, continue, or pause messages.

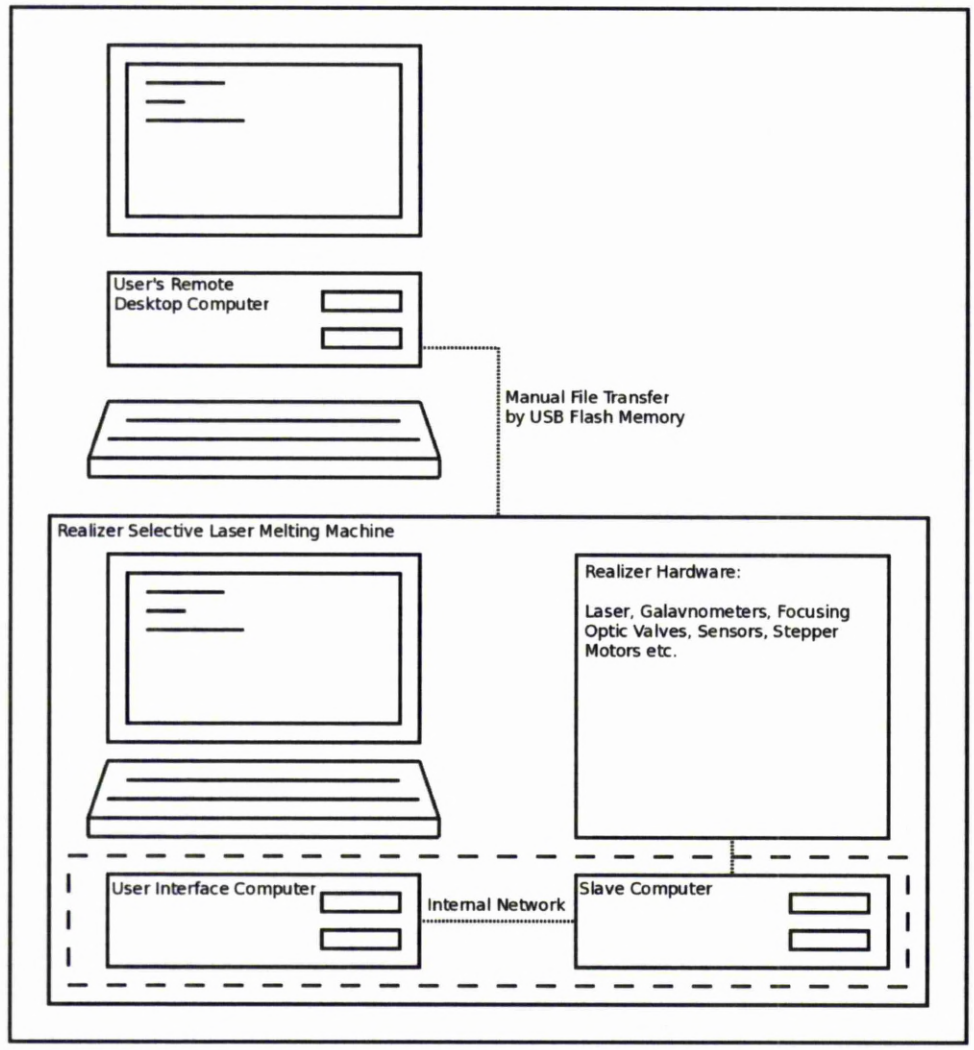


Figure 58: Computers used in building parts on the SLM equipment – identifying the computers used during the automatic build process.

The commands are passed to the slave through a network shared directory, titled command. Once the machine has carried out the command it returns the command through a network-share called response when requested by the slave. During the build process the user interface machine will compile a set of instructions for the slave to perform in order to process a layer. This - called a job file - contains instructions on how to move the scanners, fire the laser, change scan settings, and change laser settings.

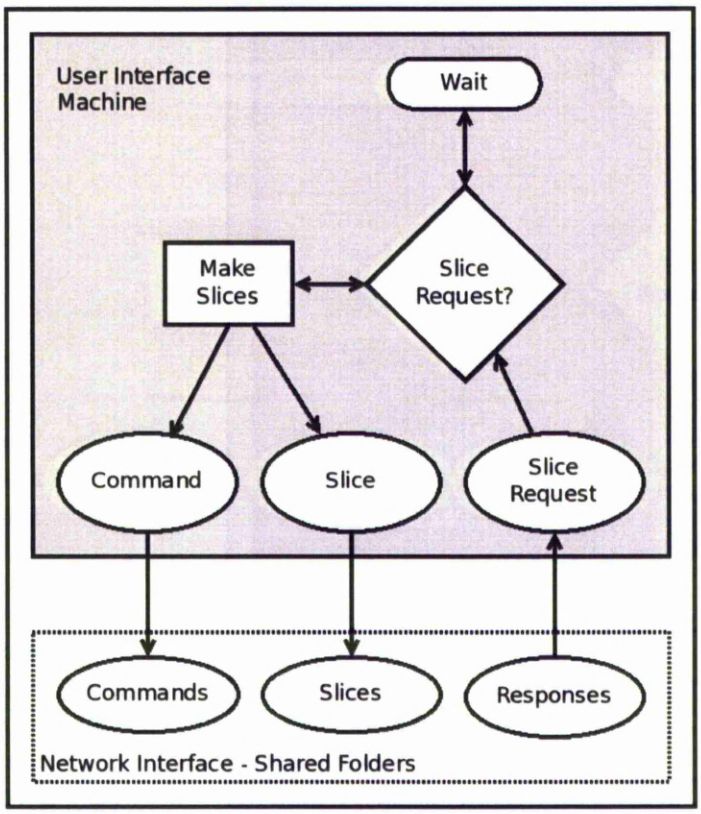


Figure 59: Flow chart of the software on the SLM user interface computer.

Once compiled this file is moved to a third network-share titled slices. When the slice file is ready the user interface machine posts a command to the slave which subsequently collects the slice and performs the instructions it contains. On completion of the slice the slave posts a request for the next slice into the response share. This process repeats until the end of the build unless an error code (for example; no powder) is raised. If this occurs the machine posts the error code through the response directory, the build pauses or stops and waits on action from the user.

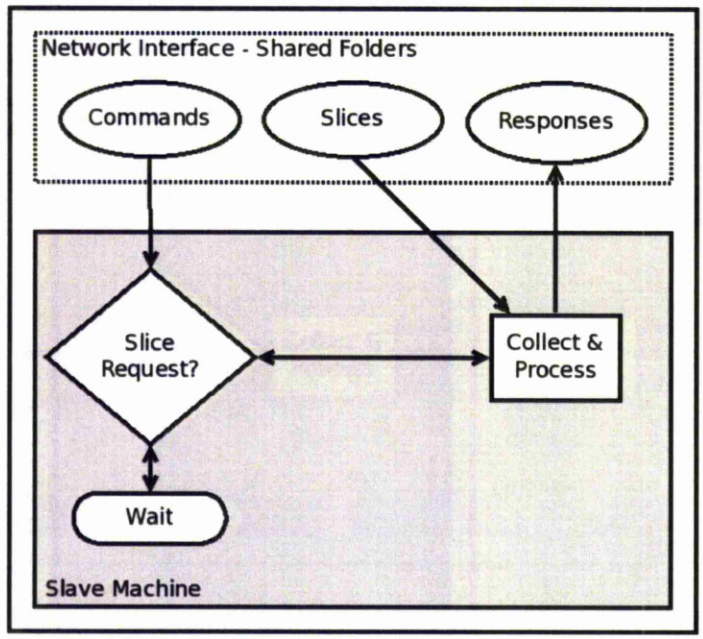


Figure 60: Flow chart of the software on the SLM slave computer.

5.1.4 Close Surface CAD Lattice Builds

It is possible to fabricate lattice structures on SLM equipment using standard techniques employed on various Layer Manufacturing (LM) machines. This process involves the slicing of a 3D triangular surface mesh file of the lattice into a sequence of 2D contours which are separated in the z-axis by a 'layer thickness' specified by the user. These structures are limited in size and or complexity due to complications relating to the processing power and memory needed to create such structures, and also minimum feature size that can be achieved using this technique.

If an end-user wished to fill a volume with a custom topology lattice structure using commercially available off the shelf CAD they must first model a single unit cell which will be repeated to create the lattice. The unit cell is then multiplied though out the cuboid bounds of the CAD volume. Finally the user will

need to carry out a boolean operation on the lattice to keep what is inside the bounding CAD and scrap what falls outside.

On completion of the lattice CAD file the data was converted into a format which the SLM build preparation software can interpret. With the SLM machine, as well as the majority of layer manufacturing machines this is the STL file format. STL format files are a mesh of triangles which form the surface of the lattice file.

As the STL files are made out of a collection of two-dimensional triangles then the resulting file is a faceted approximation of the part the end user wishes to create. If the resolution of the file is very fine, the resulting mesh of triangles is also very fine. This takes a long time to produce, and will mean the next pieces of software in the chain will also take far longer to work on the file due to its size. Ideally the resolution should not be far beyond the capability of the machine. The facets need only be small enough that they are not visible on the final part. Depending on the requirements of the part a degree of faceting on the final part may also be acceptable which would reduce the required processing time further.

The intersection of a lattice structure model (whose elements are not parallel to the slice plane) and a slice plane comprises of various ellipses and unions thereof. These ellipses are represented by a sequence of coordinates. The minimum number of points required to define a poorly-faceted approximation of these ellipses is three. Once fabricated this would result in a lattice structure with elements in one of three categories. Triangular-prismatic elements would occur if the points where all aligned. Elements with a triple helix twist would be produced if the points representing the contour progressively twisted about the centre of the element. Roughly cylindrical elements with poor surface finish (if they built at all)

would result if the points location on the ellipse where random on each subsequent layer.

Once the mesh file has been created it is loaded into build preparation software that is normally specifically designed for manipulating the triangular mesh data for RP machines, and in some cases developed specifically for the target machine. In the case of the SLM machine this is the propitiatory software supplied on it. With this software the part is positioned at the correct location on the build, supported, sliced at given layer heights, and processing parameters are assigned to the part. This process is repeated for all parts to be built on the machine.

The supporting structure is required because of the limitations in minimum angle from the horizontal that a surface can be built at before it fails to build by either breaking up, suffering extremely poor surface finish, or simply missed out entirely. The supports need to be considered carefully as too much supporting will increase the amount of time the part requires in finishing and the subsequent reduction in quality of the downward facing surfaces where the supports attached, and too little will lead to the failure of the part.

The processing parameters need to be selected from within a processing parameter window that creates parts of the required density, and surface finish. Excessive laser energy delivered to the part can lead to balling, similar to that discussed in the literature survey while too little will mean the powdered metal has only been sintered, or not joined at all rather than fully melted.

During the build process, particularly if the geometry has not been run before the end user may wish to observe the machine processing the parts to establish how

close to the ideal processing parameters they are running. If need be the user may repeat the build with adjusted settings to improve part quality.

On build completion the parts will be cut from the substrate (unless the substrate is forming an integral section of the finished part) and the supporting structure will be removed. Further finishing will depend on the required specifications for the part. This may involve sand, bead, or shot blasting, or even polishing.

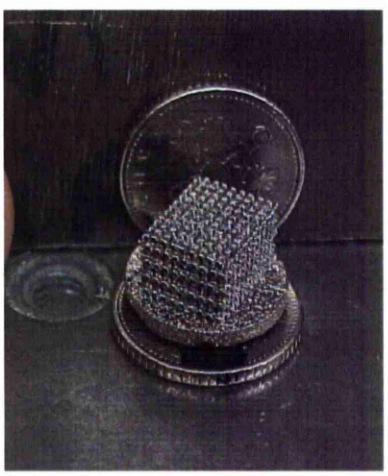


Figure 61: Lattice structure build using standard procedures and commercially available CAD software.

5.1.5 Open Surface CAD Lattice Builds

Open-surface CAD models can also be sliced, but not hatched. The intersections of the open surface and the slice planes result in contours that are not closed. Various LM process – including SLM – require support structures to be included in the slice data in order to build part with over-hanging features correctly. This support structure can be lattice-like structures that have been created with a minimum of two triangles representing each element. In the CAD model this element has no volume, as it is a flat surface. The intersections of this element and a slice plane is a straight two part line if the triangles are on the same plane, or a

two line contour defined by three points; start, middle, and end.

5.1.6 File Formats

While using the RealiZer machine the user will typically come across three main types of file. These are the build, part, and material file. Also critical to the latter parts of this research is the single slice file which is sent from the User Computer to the Slave Computer just prior to scanning a layer and contains all the information required by the Slave to scan the layer and prepare the machine for the next slice by dropping the elevator by a distance specified in the file and adding powder. There is the option to include the information contained in the material file in the header of the part file. This was never done by the lattice preparation software discussed in this thesis as a new file would be needed for each lattice slice data file that is processed with different processing parameters. Instead the user is left to assign material data when the controlling software complains that there is none associated with the file.

5.1.6.1 Build File

The build file has a simple structure which is possible to prepare away from the machine with a text editor if the user does not have access to a licensed seat of the build preparation software. For complex experiments it was found to be more convenient to create a short script to compile these files. An example of a simple build file is shown in Figure 62.

```
[MODELS]
count=4
Model0=E:\SLM-DATEN\Build files\xxxWB\big.f&s
Material0=xxxWB.dat
offsetX0=-19.327
offsetY0=5.0
offsetZ0=0.0
shiftSlices0=2
Model1=E:\SLM-DATEN\Build files\xxxWB\big_sup.f&s
Material1=xxxWB.dat
offsetX1=-19.327
offsetY1=5.0
offsetZ1=0.0
shiftSlices1=0
Model2=E:\SLM-DATEN\Build files\xxxWB\simple.f&s
Material2=xxxWB.dat
offsetX2=5.0
offsetY2=5.0
offsetZ2=0.0
shiftSlices2=5
Model3=E:\SLM-DATEN\Build files\xxxWB\simple_sup.f&s
Material3=xxxWB.dat
offsetX3=5.0
offsetY3=5.0
offsetZ3=0.0
shiftSlices3=0
```

Figure 62: A complete build file.

The header for the build file is very short, it simply comprises a count for the number of parts listed in the rest of the build file. It's critical to note that this value is a count, and not the highest model identification number. The first model of any valid build file will have a model identification of 0, and subsequent models will have consecutive numbers from there. Therefore the highest identification number used when the count is 4, is 3. This is shown in Figure 63.

```
[MODELS]
count=4
```

Figure 63: Build file header

Each part is listed in the build file with the six lines/variables as shown for an example part in Figure 64. The model variable must be the location of the part when it is ran on the machine, so if the build is prepared remotely this must be

adjusted to match the machine location, this is easily achieved with a find and replace operation.

```
Model3=E:\SLM-DATEN\Build files\xxxWB\simple_sup.f&s
Material3=xxxWB.dat
offsetX3=5.0
offsetY3=5.0
offsetZ3=0.0
shiftSlices3=0
```

Figure 64: Part reference from a build file.

5.1.6.2 Part File

Once the CAD file has been sliced, hatched, and saved the build preparation software outputs a slice part file. When this file is created in ASCII format the file is readable in text editors. The file comprises of a header file followed by the individual slices at given z-axis heights. An example header is given in Figure 65.

The header contains information relating to the file when it was saved from the build preparation software. The first line gives where the file was saved to, the second what file format it is in, the third the length of the file, the fourth the number of slices, and the fifth the bounds of the file. When reading the files the build preparation software checks whether this information is present but does not seem to use it. It is possible to cut out a number of layers from the end of an part file and it will still read, however if the header is removed it will not load.

```
#
A:\10mm Box 10degz 5mm hole hatched.f&s
4 Version
349155 FileLength
234 SliceCount
-12.1844 0.6 -2.3365 10.4479 0 11.65  bounds
7.7143e-39 6.42859e-39 9.91838e-39 1.41531e-4
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LightExposureBlock
HS
14 lenght of header text
test info text
HE
```

Figure 65: Part file header

Data for the individual slices is included in the files as shown for an example in Figure 66. The "Start Slice" line identifies the start of a slice in the file. The following four lines contain header information about the rest of the slice. The first line gives the total number of groups of commands for the scanner. For example one layer containing 2 box shaped areas being horizontally hatched would have a total block count of two, regardless of how many hatch lines are contained within one hatch area. If that file then has a single contour around the edge of each hatch block the total block count would be four. Following this is the slice number of the slice being described. This must be consecutive, 0 to begin with and each following slice increasing by a count of 1. The following two lines totals the different types of hatch/contour types used in that layer. These were not required to correctly load slice data during the course of the research. The penultimate line of the slice header gives the bounds of the following blocks of data within the slice. The final line of the slice header gives the z axis height of the slice.

The data for a single slice is ordered with all the hatch and contour blocks in one set. Whether the hatch or contours are scanned first is down to user preference.

```
Start slice
2 TotalBlockCount in slice #129
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
BlockCountPerType
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AnzahlPunkteAllerBloecke
-11.6072 -1.74047 0 9.86769  bounds
6.45 zlevel
Start hatch block
8 HatchType in hatch block #0
9 HatchCount
-9.78338 -1.5 -9.22668 -1.5 start end
-8.51778 -1.375 -9.80541 -1.375 start end
-9.82744 -1.25 -7.80887 -1.25 start end
-7.09996 -1.125 -9.84947 -1.125 start end
-9.8715 -1 -6.39105 -1 start end
-5.68214 -0.875 -9.89353 -0.875 start end
-9.91557 -0.75 -4.97323 -0.75 start end
-4.26433 -0.625 -9.9376 -0.625 start end
-9.95963 -0.5 -3.55542 -0.5 start end
Start contour block
16 ContourType in contour #0
6 PointCount
-3.63425 2.81161
-3.88475 2.4539
-4.19337 2.14528
-4.55108 1.89478
-4.94687 1.71041
-5.36872 1.59719
End slice
```

Figure 66: Slice from a part file.

Within the slice data for a single slice there are blocks of hatch and contour information. The header for a hatch block is given in Figure 67. “Start hatch block” marks the start of a hatch block in the file, “8 HatchType in hatch block #0” indicates that these hatches are of hatch type 8 which is described as “Horizontal hatch” in the build preparation software and are the first hatch block in that slice, “132 HatchCount” is a total of the number of hatch lines within that hatch block. The block count continues until the contour block start, when it will be zeroed, or until the end of the slice.

```
Start hatch block
8 HatchType in hatch block #0
132 HatchCount
```

Figure 67: Hatch block header.

Figure 68 describes a horizontal scan vector of length 12mm travelling from a start co-ordinate of (12, 15) to a finish co-ordinate of (24, 15).

The command for the contours Are given after a header such as:

```
(Start x value) (Start y value) (End x value) (End y value) start
end
E.g.: 12.0 15.0 24.0 15.0 start end
```

Figure 68: Description of a hatch line entry in a hatch block.

The header for a contour block is detailed in Figure 69. "Start contour block" marks the start of a list of instructions for the scanner for one contour block. "1 ContourType in contour #3" Describes this contour bock as Contour type 1, which is Outer contour and #3 which describes this as the 3rd contour block. The 5 point count means there are five points within this contour block.

```
Start contour block
1 ContourType in contour #3
5 PointCount
```

Figure 69: Contour block header.

Figure 70 is a list of x-y co-ordinates describing a path for the scanners to take with the last and the first point being the same. The example given would draw a 10mm square with the lowest point being at (0,0).

```
(1st Point x value) (1st Point y value)
(2nd Point x value) (2nd Point y value)
(3rd Point x value) (3rd Point y value)
(4th Point x value) (4th Point y value)
(5th Point x value) (5th Point y value)

eg:
0.0 0.0
0.0 10.0
10.0 10.0
10.0 0.0
0.0 0.0
```

Figure 70: Description of a five point contour in a contour block.

5.1.6.3 Material File

The material files contain information about how the associated part files will be sliced and a number of process parameter sets describing how they will be processed by the laser. Table 2 Details the names of the parameters that can be adjusted for each parameter set.

`expo`	Exposure Time
`pdist`	Point Distance
`laserPower`	Laser Power
`laserFrequency`	Laser Frequency (legacy feature - disabled on experimental SLM system)
`lensPos`	Focus optic position
`numberOfExposures`	Number of times to repeat scan slice geometries.

Table 2: Description of the parameters that are available to adjust for each parameter set.

The exposure point is the amount of time the laser spends aimed at each point along a line. Each point is separated by a point distance. The power and focus of the laser can be adjusted for each process parameter set. The laser frequency must be a legacy feature as it was not active on the SLM equipment used for this research.

The available process parameter sets are listed in Table 3. Since there are 10 process parameter sets and 18 geometry types some geometry types share the same process parameter set. This means it is not possible to associate different process parameters for each geometry type.

`Boundary`	Contour over powder
`BoundarySolid`	Contour over pre-processed material
`Hatch`	Hatch over powder
`HatchSolid`	Hatch over pre-processed material
`SkinHatch`	Hatch on up facing surface
`FillContour`	Area fill contour
`FillContourSolid`	Area fill contour over pre-processed material
`Support`	Support geometries
`InnerSupport`	Support geometries 2
`PointSequence`	Discrete point scanning

Table 3: Description of the available parameter sets.

The material file in Figure 71 shows how the process parameter information is laid out. This information was edited using a text editor. The user could do this directly via a file manager, or if the material set is edited from inside the build preparation software it will open up a text editor. This data was associated with a part before it was sliced when using the proprietary build preparation software.

```
[50uLAYER]

exposureSequence=1

expoBoundary=200
expoBoundarySolid=600
expoHatch=300
expoHatchSolid=400
expoSkinHatch=200
expoFillContour=200
expoFillContourSolid=500
expoSupport=2500
expoInnerSupport=800
expoPointSequence=1000

pdistBoundary=50
pdistBoundarySolid=60
pdistHatch=85
pdistHatchSolid=65
pdistSkinHatch=75
pdistFillContour=50
pdistFillContourSolid=50
pdistSupport=75
pdistInnerSupport=50

laserPowerBoundary=1300
laserPowerBoundarySolid=1700
laserPowerHatch=1300
laserPowerHatchSolid=1700
laserPowerSkinHatch=1300
laserPowerFillContour=1300
laserPowerFillContourSolid=1500
laserPowerSupport=1500
laserPowerInnerSupport=800
laserPowerPointSequence=2250

lensPosBoundary=3550
lensPosBoundarySolid=3550
lensPosHatch=3550
lensPosHatchSolid=3550
lensPosSkinHatch=3550
lensPosFillContour=3550
lensPosFillContourSolid=3550
lensPosSupport=3550
lensPosInnerSupport=3550
lensPosPointSequence=3000
```

Figure 71: Example material file.

5.1.6.3.1 Material File Correlation

There are 23 geometry types that the build preparation software recognises as detailed in Table 4. Not all of these get output to the scanners, and some share process parameter sets. There are only 10 process parameter sets, each limited to either hatch, contour, or a point sequence.

Geometry Type Number	Class	Descriptive Name	Material Parameter
0	Contour	Inner Contour	BoundarySolid
1	Contour	Outer Contour	BoundarySolid
2	Contour	Inner Support Contour	Support
3	Contour	Outer Support Contour	Support
4	Contour	Offset Fill Lines	FillContourSolid
5	Hatch	Vertical Hatch	HatchSolid
6	Hatch	Vertical Support Hatch	Support
7	Hatch	Vertical Shell Support Hatch	InnerSupport
8	Hatch	Horizontal Hatch	HatchSolid
9	Hatch	Horizontal Support Hatch	Support
10	Hatch	Horizontal Shell Support Hatch	InnerSupport
11	Contour	Inner Contour on Powder	Boundary
12	Contour	Outer Contour On Powder	Boundary
13	Contour	Offset Fill Lines on Powder	Fill Contour
14	Hatch	Vertical Hatch on Powder	Hatch
15	Hatch	Horizontal Hatch on Powder	Hatch
16	Contour	Original Inner Contour	(Not Output)
17	Contour	Original Outer Contour	(Not Output)
18	Contour	Original Inner Contour on Powder	(Not Output)
19	Contour	Original Outer Contour on Powder	(Not Output)
20	Contour	Point Sequence	PointSequence
21	N/A	Unused	N/A
22	N/A	Unused	N/A
23	Hatch	Vertical Skin Hatch	SkinHatch
24	Hatch	Horizontal Skin Hatch	SkinHatch
25	N/A	Unused	N/A
26	N/A	Unused	N/A
27	N/A	Unused	N/A
28	N/A	Unused	N/A
29	N/A	Unused	N/A
30	Hatch	?	(Not Output)

Table 4: Table linking the slice file geometry types to process parameter sets

The information in Table 4 was compiled by manually creating a single slice ASCII format part file and for each geometry type trying contour, or hatch. There is an option on the build preparation software to remove a geometry type from a

part file during build preparation. If a geometry type isn't present the option to delete this geometry type is greyed out. Therefore once the information was correctly loaded into the build preparation software the geometry type could be identified.

Material Data File Parameter	Types Influenced
Boundary	12, 11
BoundarySolid	0, 1
PointSequence	20
Hatch	14, 15
HatchSolid	5, 8
SkinHatch	23, 24
FillContour	13
FillContourSolid	4
Support	2, 3, 6, 9
InnerSupport	7, 10

Table 5: Table detailing which slice file geometry types are controlled by the same process parameter set.

To confirm the link of geometry types to their process parameter sets a material file was created with unique values for the laser power for each process parameter set. Once this was done the slave computer was turned off and the build process started on the user interface machine. The appropriate responses were created and placed into the response directory and the user interface computer created the single slice file that would have been loaded by the slave machine had it been running. This was interpreted to find out which laser power had been applied to each geometry type, and thus confirming the used process parameter set. The links between process parameter sets and geometry types are shown in Table 5.

5.1.6.4 Single Slice File Format

The slice file is created by the User Interface Computer and contains all the information to scan a layer and how much to drop the elevator by prior to the recoating process. This is created by the user computer when requested by the

slave computer via the response directory.

Once the slice file is created in the slices directory a command is sent to the machine through the commands directory to copy and carry out the commands within the slice file. The machine responds through the responses directory when it has copied the file and started to process it.

Block	Commands	Description	Number Split (x)	Conversion
Header	L0000000500	Layer Number, Layer Height	00001, 00500	x, x / 10^4
Start Block	E0000000005	Start Hatch Block		
Block 0	R5000350003	Precision (P)	50003, 50003	x - 50000
	Z0000000000			
	O4875048750	Origin	48750, 48750	(x -50000) / 10^1
	P0200000000	Laser Power	0200000000	x / 10^8
	F0150001500	Laser Focus	01500, 01500	x / 10^2
	T0250000205	Exposure, Point Distance, Geometry Type	02500, 002, 05	x, x * 10, x
	M5500050000	Move to Point	55000, 50000	(x – 50000) / 10^P
	E0000000010	Delay		
	D5500070000	Draw to Point	55000, 70000	(x – 50000) / 10^P
	B0000000000	Possible pause point		
Block 1	R5000350003	Precision	50003, 50003	x - 50000
	Z0000000000			
	O4895048750	Origin	48950, 48750	(x -50000) / 10^1
	M5500050000	Move to Point	55000, 50000	(x – 50000) / 10^P
	E0000000010	Delay		
	D5500090000	Draw to Point	55000, 90000	(x – 50000) / 10^P
	B0000000000	Possible pause point		
Block 2	R5000250002	Precision	50002, 50002	x - 50000
	Z0000000000			
	O4915048750	Origin	49150, 48750	(x -50000) / 10^1
	M5050050000	Move to Point	50500, 50000	(x – 50000) / 10^P
	E0000000010	Delay		
	D5050056000	Draw to Point	50500, 56000	(x – 50000) / 10^P
	B0000000000	Possible pause point		
Start Block	E0000000003	Start Contour Block		
Block 3	R5000250002	Precision	50002, 50002	x - 50000
	Z0000000000			
	O5000050000	Origin	50000, 50000	(x -50000) / 10^1
	P0100000000	Laser Power	0100000000	x / 108
	T0050000601	Exposure, Point Distance, Geometry Type	00500, 006, 01	x, x * 10, x

	M5000050000	Move to Point	50000, 50000	$(x - 50000) / 10^P$
	D5000051000	Draw to Point	50000, 51000	$(x - 50000) / 10^P$
	D5015051000	Draw to Point	50150, 51000	$(x - 50000) / 10^P$
	D5015050000	Draw to Point	50150, 50000	$(x - 50000) / 10^P$
	E0000000010	Delay		
	D5000050000	Draw to Point	50000, 50000	$(x - 50000) / 10^P$
	B0000000000	Possible pause point		
Footer	S0000000000			
	C0000100000			
	W5000076000	Wiper end points	50000, 76000	$(x - 50000) / 10^2$
	E0000000000	End of layer		

Notes:

- 'x' in the conversion box refers to the number retrieved from the command
- When the command is broken into two numbers if either part of the split needs a different conversion the conversions are split by a comma. If the comma is not present the conversion applies to both halves.
- The superscript 'P' used on the 'Move to' and 'Draw to' commands refers to the precision found from the previous Precision command from the job.dat command.
- L9999900000 as a layer command tells the machine that this is the last layer and the machine will shut down after completing the layer.

Table 6: Example job file broke into sections.

Table 6 describes the job.dat file in enough detail to be able to create job.dat files that will run from the machine. The conversion column converts the values into either millimetre in the case of origin, move to, and draw to commands or to the same units as in the material files.

5.2 Methodology Summary

Commercially available equipment provided a way to create lattice structures in both open and closed surfaces, but serious limitations were encountered. When attempts were made to use standard techniques to scale the lattice structures from a scale seen in Figure 61 to larger size blocks using a similar density of lattice.

The first aim of the research was to identify and investigate limitations of the current equipment and the initial limitations are more than clear. Up scaling of computer resource and migration onto distributed computing resources may have improved matters but this may not be cost effective. Investigation was required to

ensure the data being created is suitable for the machine and structures, and the best use is being made of standard computer resources before investing in distributed systems.

The ratio of metal volume to total space the lattice occupies (relative density of the lattice) is high in structures like that shown in Figure 61. Reducing this by ensuring the scan strategies for the lattice structures create the smallest diameter trusses is another clear aim highlighted by the standard techniques for lattice generation.

6 Results and Discussion

This work focuses on reducing the perceived limitations of the SLM process in creating lattice structures. The two main focuses of the research are; data preparation, and increasing the range of lattice structures that can be built on the RealiZer SLM machine. There were many issues to tackle relating to data preparation including ease of creating the required files to run the SLM machine, the consistency of the slice data, and enabling the creation of new geometry types.

The limitations to which lattice structures could be built on the machine were reduced through development of data preparation techniques and development of the the machine software. Adjusting standard user variable processing parameters improved the formation of lattice structures using the initially developed techniques. Far more success was achieved with the creation of new machine control software that enabled new scanning techniques. These techniques were aimed at better control the melt pool.

6.1 Research into Creating Lattice Part Files

The creation of the slice data that is needed to create lattice structures on the SLM machine is challenging due to the inherent complexity of the lattice structures. This is exacerbated by a requirement to reduce the cell size and element diameter of the lattice as far as possible. The greater the range of cell sizes and range of available solid/pore volume ratios the greater the range of applications the structures could be utilised for.

The research carried out led the development of software for more efficient methods of creating fault free CAD data, through various techniques of modelling

the lattice data as elements and through to replacing the proportion of the machine control software that resides on the User Computer with a solution optimised for lattice systems.

6.1.1 Standard Surface Lattice Builds

The initial research focused on a more efficient way to create lattice CAD files which were subsequently sliced and hatched by the standard build preparation software. This software worked by creating a triangulated surface a set distance from a given lattice element model that used implicit and marching cubes tools from an an open source data manipulation and visualisation tool kit, VTK [41].

A element model of a lattice is one where the links are modelled merely as an array of points, and a list of links which are each two integer numbers that pointing to coordinates in an array.

Figure 72 shows a single unit cell of a lattice structure with octahedral links and vertical pillars. This and Figure 73 were created with a resolution that is half that of the element radius. This gave a good surface finish to the model.

Figure 72: Computer rendering of a single octahedral and pillars lattice cell topology.

The computer model shown in Figure 73 was used to create a test part for an Envisiontec machine which is a different layer manufacturing technology based on light projection exposing whole layers with UV light on photo polymeric resin.

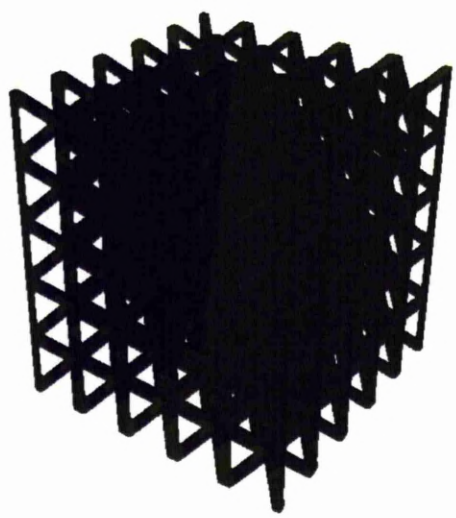


Figure 73: Computer rendering of a 10mm side cube block of octahedral and pillars lattice.

The model was created by exporting the native triangular surface mesh model into an STL binary file format. The file size of this part was large at 23 megabytes.

Parts of higher complexity would fail to process on this machine, and parts with a smaller cell size are beyond the capabilities of the system. Figure 74 shows the part that was created, placed on a Sterling pound coin.

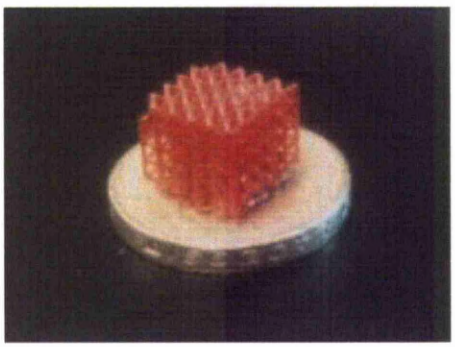


Figure 74: Lattice structure created on a photopolymeric resin based system.

Figure 75 through to 78 show some tests that where carried out to find the lowest resolution that could be used to create the lattice structure surface model. Figure 75 shows the same ratio between element radius and resolution as used for Figure 73 which results in a very smooth model, but as previously discussed it was a high file size.

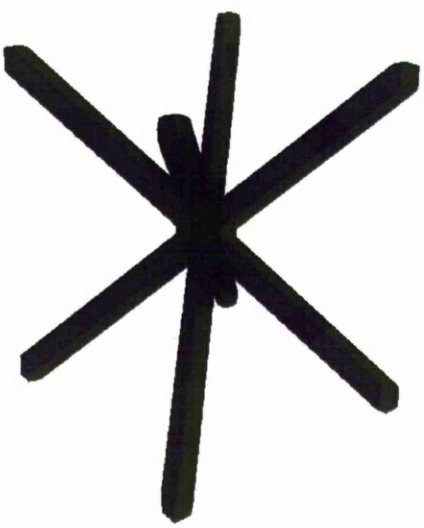


Figure 75: Computer rendering of a 5mm cube lattice with an element radius of 0.25mm and a resolution of 0.125mm

Figures 76 and 77 show increasing faceting, which is shown by the triangles that form the surface model becoming prominent and easily noticeable.

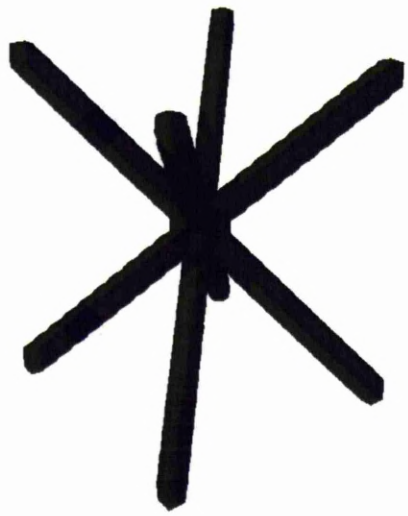


Figure 76: Computer rendering of a 5mm cube lattice with an element radius of 0.25mm and a resolution of 0.188mm

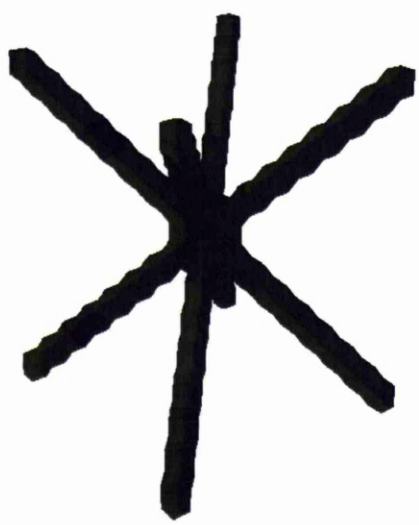


Figure 77: Computer rendering of a 5mm cube lattice with an element radius of 0.25mm and a resolution of 0.250mm

Figure 78 Shows a model that would not build the desired lattice geometry as the surface has broken up into many disconnected shells, rather than one continuous

surface.

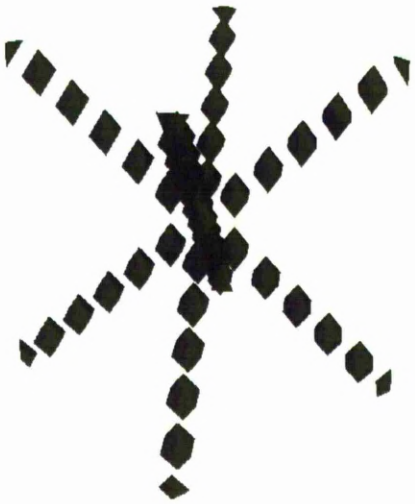


Figure 78: Computer rendering of a 5mm cube lattice with an element radius of 0.25mm and a resolution of 0.313mm

Figure 79 shows the relationship between the file size of an exported STL file and the resolution of the facet surface generation. For reference the file size of the element model used as a skeleton to form these surface models is also included. The four triangular mesh files (labelled by their resolution) were output in a binary format while the element model was output in a readable ASCII format, but could be output in a binary STL format to further reduce file size.

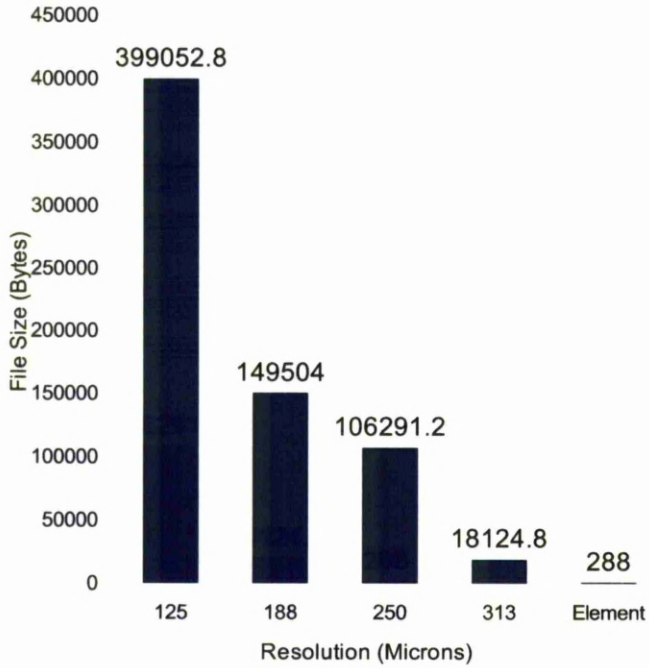


Figure 79: Model resolution plotted against file size

A small readable ASCII version of this file format is illustrated in Figure 80, which defines a unit structure with four vertical pillars. Creating the lattice in this manner has the benefit that it is very robust, and has no redundant data. If for example the lattice was stored as a sequence of double co-ordinates with no shared points, merely relying on the points being coincident then gaps in the structure may be introduced into the structure by rounding errors.

```
POINTS 8 float
0 0 0
1 0 0
1 1 0
0 1 0
0 0 1
1 0 1
1 1 1
0 1 1

LINES 4
0 4
1 5
2 6
3 7
```

Figure 80: Example file describing four vertical pillars.

6.1.2 Existing Open Surface Lattice Structures

Lattice structures were already used in layer manufacturing technology as a supporting technique. This system provides a disposable structure that allows easier removal of the part from the substrate whilst still providing anchor points from which the part can be built on. Slice data for these support structures are created by the build preparation software slicing open surface truss elements. A section from a layer of a slice data created in this manner is shown in Figure 81. The lattice structure that was used as a commissioning part for the RealiZer machine was one of these files.

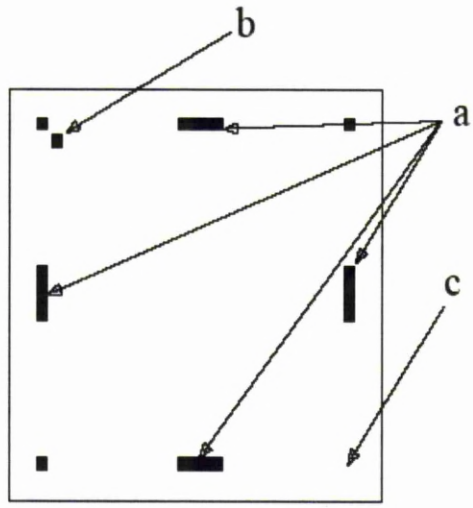


Figure 81: Slice data from supplied lattice support file.

Figure 81 shows the slice data of a single cell within an open surface support structure. The irregularity of the file is shown by double points at location b, missing points at location c, and a mix of long (a) and short scan vectors. The short vectors were intersection data from the centre sections of the trusses. The long vectors occurred where the open surfaces for two or more trusses joined. This joining required extra triangles, and the combination of the triangles forming the two or more trusses, the additional joining triangles, and automatic mesh and slice data cleaning by the propitiatory software resulted in the long vectors.

The short vectors only received a single exposure from the laser because they were less than one point distance in length. The longer received many overlapping exposures. This resulted in a very small processing parameter window. Each of the overlapping exposures reheated a hot possibly stil molten volume of material and melted some more power. Each exposure point of the single exposures only affected virgin powder. The exposure time needed to be large for the points that formed short vectors, but small enough to avoid processing the longer vectors.

The apparent irregularity of the file is present because the exact control of how the 3D open surface and 2D slice file where constructed and the slice data cleaned up. Due to the software being closed source, and there being no facility to adjust these steps it was not entirely clear what was being done to create the 3D and 2D data. There was a need to create slice data without these errors for it to build correctly on the SLM equipment. In order to progress the research from this point a new method to produce the slice data was required.

6.1.3 Direct to Slice File Lattice Structures

The number of points required to make a simple 4 by 12 by 6 cell structure made attempts to output the point data using spread sheet software impractical. The most logical path from this point was programming bespoke software to create the slice data directly. A small program was provided based on these requirements to create a simple fixed cell geometry lattice structure.

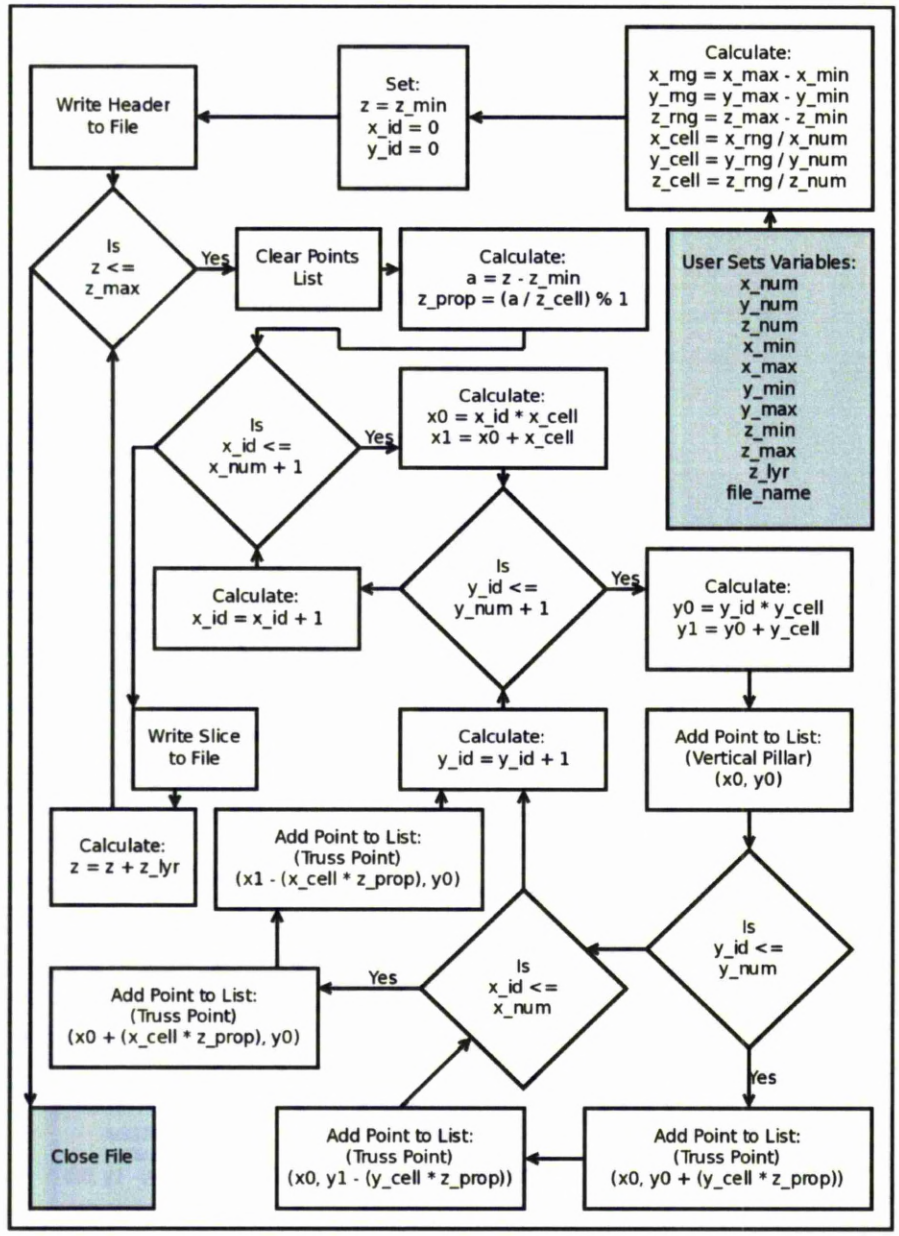


Figure 82: Flow chart of program to create lattice structures.

The written program (detailed in Figure 82) created the slice files for the lattice structure directly without the need to model the lattice structure in a CAD package. The user started by modifying the program's script. By doing this the user could change the size of the bounding cuboid that contains the lattice, the number of cells in each axis, the layer thickness, and the output file name.

This set up sets:

- [x/y/z]_num is the number of cells in the [x/y/z]-axis.
- [x/y/z]_min is the minimum extent of the lattice in the [x/y/z]-axis.
- [x/y/z]_max is the maximum extent of the lattice in the [x/y/z]-axis.
- z_lyr gives the slice thickness.
- and file_name defines the file name.

During the course of the program:

- [x/y/z]_id gives the current cell number in the [x/y/z]-axis.
- z gives the current z level.
- z_prop gives the proportion of the way through the current z cell.
- [x/y]0 gives the lowest [x/y] value of the current cell.
- [x/y]1 gives the maximum [x/y] value of the current cell.

This reduced the time required to create the slice files significantly. Previously the available CAD software had struggled with the complexity of the structures, and used significant computing resource (processing time, memory usage). The lattice structures where defined by extents, and number of repeating cells in each axis. The slice files where created by calculating the intersection of the regular repeating cell with the slice plane by evaluating the z-planes position in each cell as it progresses up through the structure.

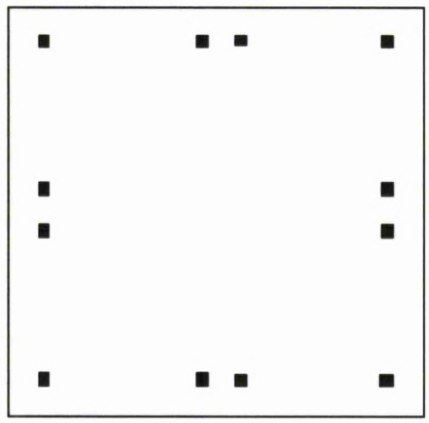


Figure 83: Recreated slice file from direct to slice software.

The effect of the software on the quality of the slice data can be clearly demonstrated by comparing the results shown in Figure 81 and Figure 83.

There are applications for rectangular blocks of lattice structure, such as sandwich panels and impact absorbing structures. However, shaped lattice parts can be made by utilising LMs ability to create free-form geometries. Research and development was focused on achieving this and the software was developed to include the ability to produce these shaped lattice structures. The clipping process shown in Figure 84 was developed as an add on to the first program. In this instance a lattice structure is created to fill the full extents of the CAD file, which the lattice structure is to be shaped to. The lattice data is still created slice by slice to fill the entire extents of the CAD, then each slice of the lattice intersection points was compared to a slice contour of the CAD file at the same level. The lattice intersections that are inside the CAD volume are kept, those that are not are rejected from the the slice file. The processes used to shape the lattice data changed with the progression of the software and are discussed in detail in a later section of this thesis.

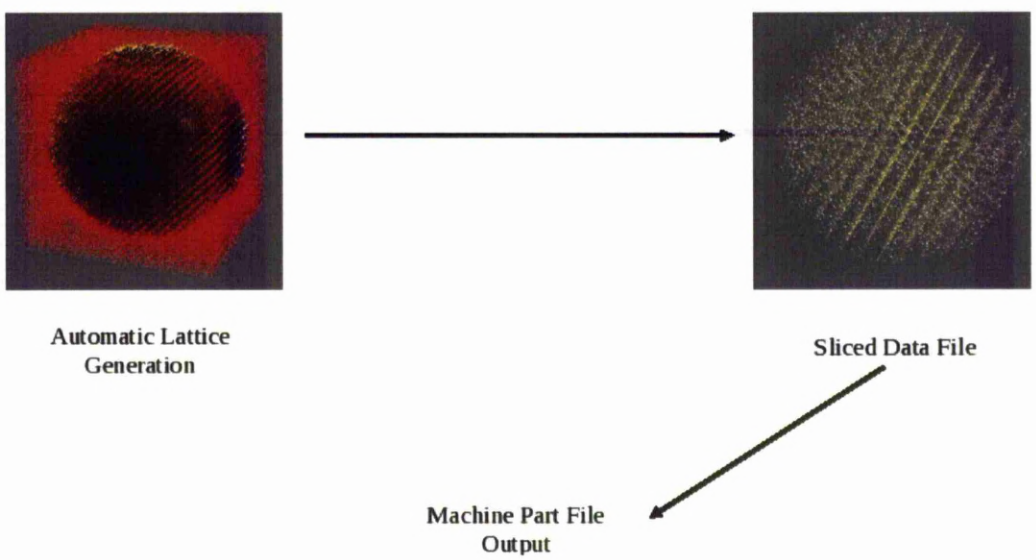


Figure 84: Software lattice generation process.

6.1.4 Comparisons of Lattice Element Representations

Figure 85 shows three methods of modeling the links in a lattice structure. The most complex is the closed surface. Each truss in the lattice structure would require 6 co-ordinates, and 6 triangle ID lists, each comprising of three integer numbers pointing to co-ordinates in the array. This is without closing the ends of the elements – which would only be required on elements that aren't connected to other elements on both ends. Each integer would be stored as at least a 4 byte value, and the same for each decimal value. This means that the minimum space required per link would be 144 bytes. The open surface link would use 72 bytes, and the element model with a file structure similar to that shown in Figure 80 would require 32 bytes.

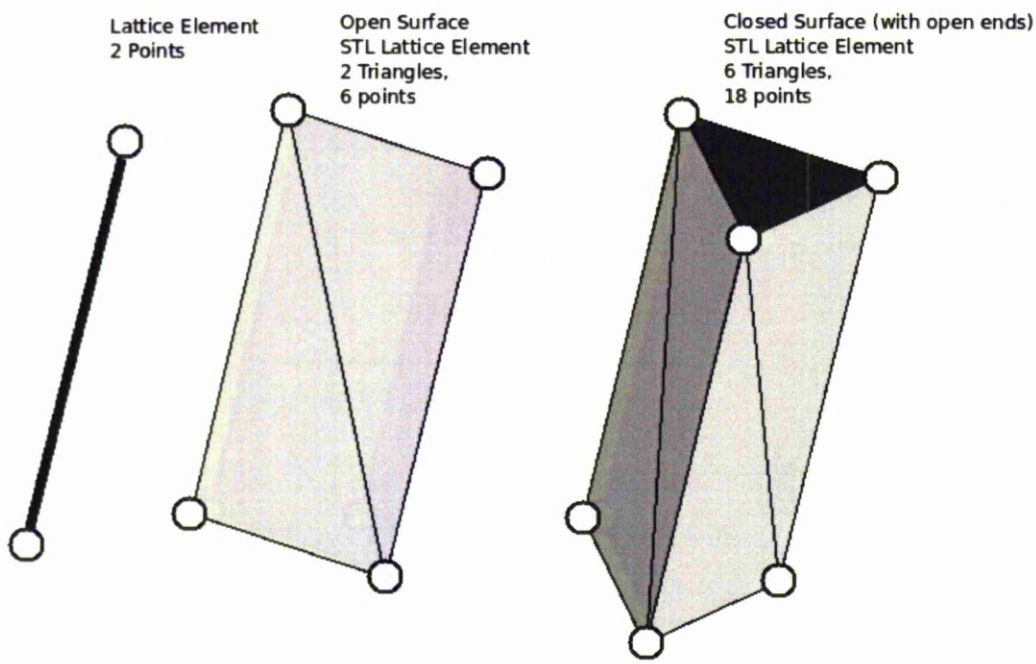


Figure 85: Three approaches to modelling segments of lattice structures.

A full lattice structure would not merely be a multiplication of the values in the previous paragraph. There are factors that would make it less, such as points in the array shared between multiple links. Other factors which would make it larger

such as an allowance for headers, footers, and formatting in the file. While joining multiple links at one location would be no problem for the element model it would cause greater increases in file size for the open and closed surface models as the mesh would have to be continuous around the elements without unwanted gaps. While the discussed method of generating an implicit surface a fixed distance from an elemental model of the lattice has no problems dealing with these joins (as the resolution is high enough), the situation may be difficult to handle in standard CAD packages and require a high degree of expertise from the CAD operator. The easiest option for the CAD operator is to raise the resolution to the limits of the application, hardware, and or the available processing time. While this may result in a very smooth looking STL file the larger the file the longer it will take to prepare for the machine.

There are also size reductions when slicing the element style of lattice model over the triangular mesh. Figure 86 shows examples of what would be output from the slicing of the element model, open surface model, and closed surface models of the lattice structure.

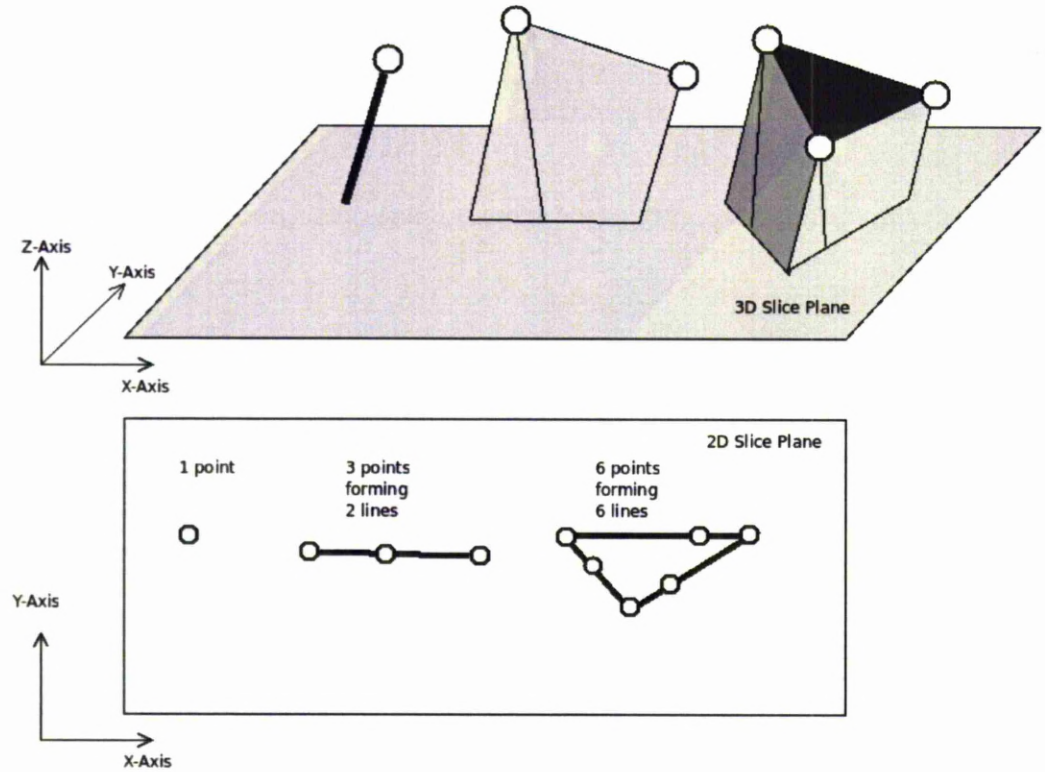


Figure 86: 2D slices of the three approaches to modelling segments of lattice structures.

Figure 86 shows a reduction in the amount of data in the slice for the element over the open surface, and the open surface over the surface modelled truss. Modelling the lattice element as a line gives a single point intersection between it and the slice place. This intersection is the simplest to describe, and requires the least data to describe. A single point intersection is a two component co-ordinate (as the third – z height is common to all points in the slice it would be recorded in the slice header) of a minimum size of 8 bytes. The open surface intersection would be three co-ordinates plus totalling 12 bytes, and the closed surface would be 6 points and 48 bytes.

The number of bytes given in the examples are for demonstration purposes, and will vary depending on the way the software is programmed, operating system,

and whether the processor is 32, 64, or more bits. The relative sizes of each will remain comparable to the examples.

6.1.5 Processing Using 3D Geometry

As the bounding lattice geometries became denser in cell size and larger in overall extent creating the slice data directly without 3D geometry became less efficient. Figure 87 Visualises processing of the lattice to identify links that;

- have no intersection with the CAD data and are outside,
- have no intersection and are inside the CAD,
- that intersect the CAD but have the lower point outside,
- and who intersect the CAD but have the upper point outside.

There is a distinction made between the elements whose lower or upper point is outside the CAD as the links with the lower point outside will definitely fail to be built unsupported as they are not connected to anything.

The details of clipping the lattice elements to the right size, supporting the lattice structures, and lattice structures built on non-rectangular grids are detailed later in this section.

Figure 87: Lattice geometry created as a visualisation for a magazine article.

6.1.5.1 Random Structures

While the strength of the SLM technology lies in its ability to create regular designed structures successfully building a random structure would demonstrate further the process capability. Beyond this the ability to create a random structure could be used as part of a procedure to create an optimised structure. While not followed through to building parts for reasons detailed later it was intended to fill a CAD geometry randomly with points. As the CAD geometry was filled the proportional chance that a point would be accepted into the final lattice depended on the results of an FEA analysis of the bounding CAD. There would be a greater probability of acceptance where the higher stress levels exist. This would lead to a dense lattice in areas of high stress, and lower in low stress region. While this structure wouldn't be controlling how elements are stressed this could be in intial step in an iterative process aimed at creating an optimised structure. The orientation of the elements may be aided if the space within the CAD geometry was distorted before being filled with points according to the direction and inverse of the magnitudes of strains in the FEA results and stretched back out after meshing. However, it would be completely impossible to generate the slice data

for random structures on a slice by slice basis. Figure 88 details a structure created in this manner that has been subsequently surfaced for visualisation purposes.

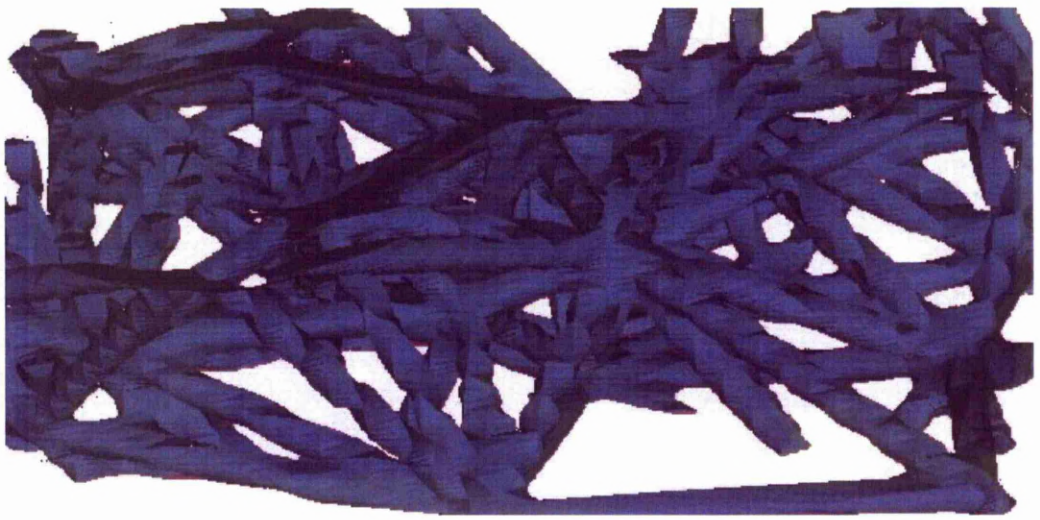


Figure 88: A surface modelled random lattice

Figure 89 shows a more complete random lattice structure. Due to the points being parsed to ensure the lattice data is contained within the CAD a secondary lattice shaping operation isn't needed as much, and less so the denser the lattice. The lattice shown here is not meshed, but some elements are going from inside to outside and back inside of the CAD data regardless of all the points being contained within the CAD. This is more pronounced in sharp angled internal corners in the CAD data.

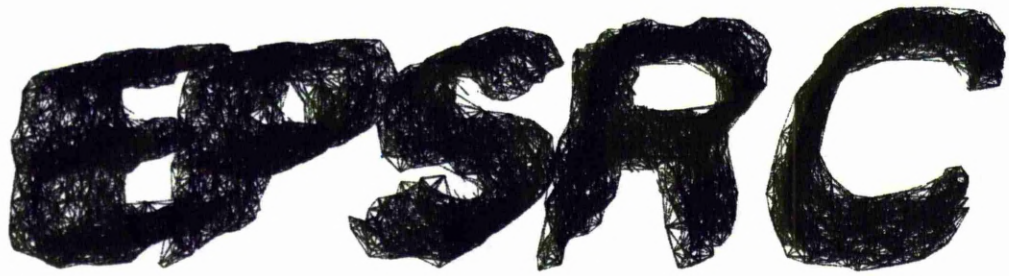


Figure 89: Randomly meshed visualisation for an EPSRC project meeting

6.1.5.2 Polar Mapped Structures

The regular lattice structures were built upon a regular grid of points. Early collaborative work investigating the application of lattice structures in orthopaedic implants identified that surgeons did not like the regular patterns that occurred on the surface of the lattice structures when automatically cut back to curved surfaces by the lattice generation software prior to being built.

Initial investigations centred on building the elements upon a polar grid rather than a rectangular grid. A visualisation of this structure is shown in Figure 90. Densification of the lattice can be observed towards the pole of the lattice structure. The effect of this on the resulting built part quality is discussed later.

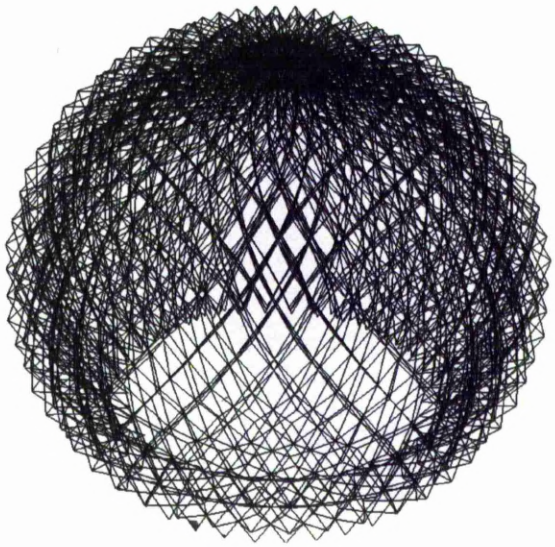


Figure 90: Standard octahedral geometry mapped onto polar rather than grid coordinates.

6.1.5.3 Randomisation

Randomisation is where the intersection points of the regular lattice structures are moved a random distance within a specified range from their original location. With this feature it was no longer possible to calculate where the intersections between the slice plane and the elements would be. Due to the random nature of the point displacements the generation of 3D geometry was required before slicing. To enable the randomisation it was necessary to create the whole lattice structure oversized to ensure the whole bounding CAD was filled. The amount the lattice structure was oversized prior to clipping was dependent on the level of randomisation. The control values for the generation of the random displacement were minimum and maximum displacements relative to the cell size in the x, y and z axis.

Figure 91 illustrates the area that a point could be moved to with a minimum and maximum randomisation range specified, and the subsequent effect that a moving

the point to a location within the range would have to the lattice structure.

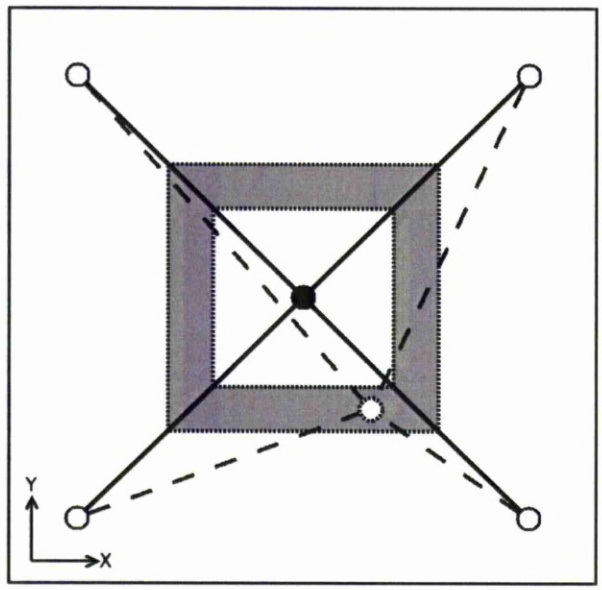


Figure 91: Illustration of the area a point could be moved to with a minimum move of 0.4 and a maximum of 0.6 in X and Y. A moved point added to illustrate the effect of randomisation.

Figure 92 shows a regular structure that has been partially randomised. This has created a slightly untidy appearance to the lattice. However, using appropriate control values for the randomisation was enough to remove the patterns that appeared on the surface of parts and satisfied the surgeons concerns over the surface patterns on the original structures.

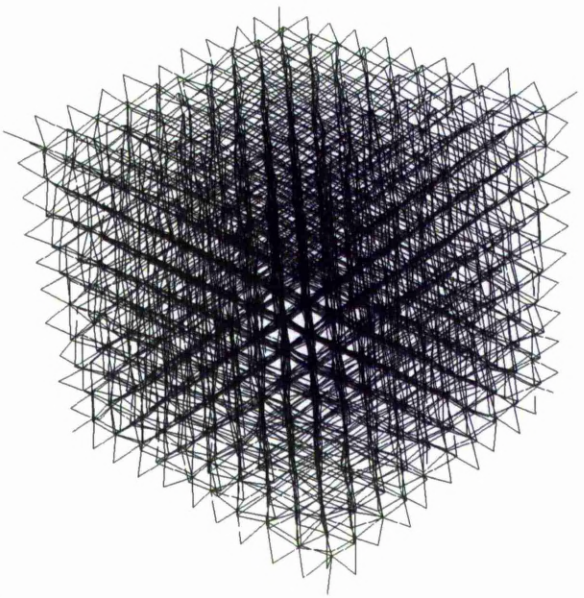


Figure 92: Regular structure whose elements have been randomly displaced within specified minimum and maximum movements.

A problem encountered early on in the development of these randomised structures was their attachment to the substrate on which they were built. The points in the lattice structure that moved below this plane were not a problem as the points below the substrate are not output to the slice data, and the resulting elements are still connected to the base plane. The points that moved off the plane away from the substrate rather than into it created unsupported sections of lattice. The structures often recovered due to the self supporting nature of the lattice as the build hight increased. Figure 93 Gives a 2D illustration of how the a structure can recover after a proportion of points do not adhere to the substrate.

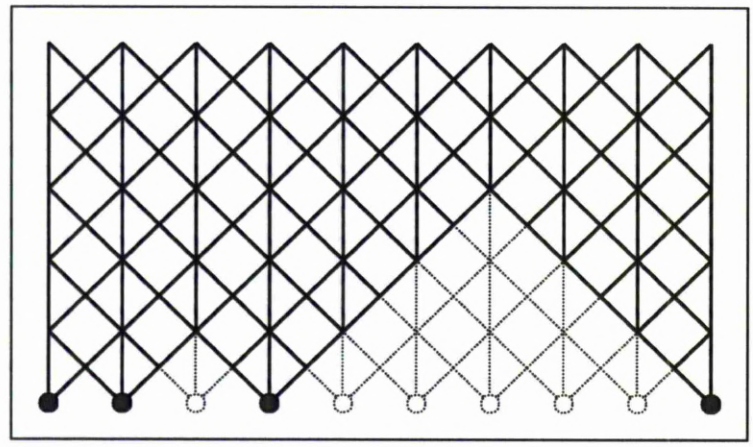


Figure 93: Self supporting nature of the lattice structures. Detailing the recovery after 6 (unfilled) out of 10 points did not adhere to the substrate.

Methods were investigated into how to better support these structures. The first of these is shown in Figure 94. Here all the points within a specified distance of the base plane are not randomised, resulting in an even connection to the base plane as shown.

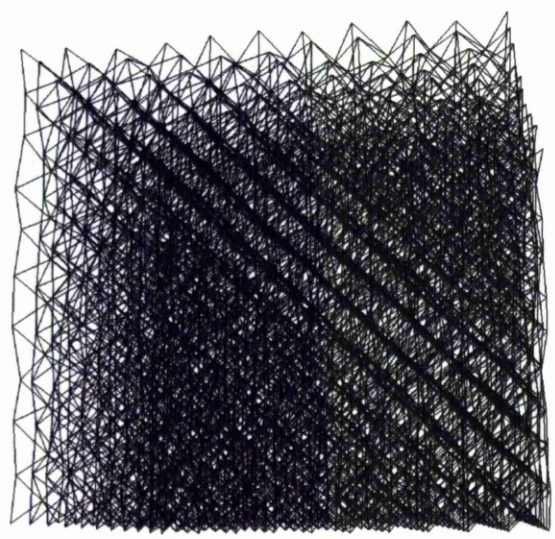


Figure 94: Randomly disrupted regular structure whose base points remain non-displaced to provide a suitable connection to the substrate

6.1.5.4 Supporting Strategies

Similarly to the problems associated with the base points moving off the substrate

plane during randomisation a lot of shaped lattice structures required a degree of supporting. The simplest and most commonly employed was the repetition of the intersection data for the first slice of the data for a predetermined height before progressing with the part. The supporting structure was saved separately to enable the user to tailor the processing parameters to best suit the needs of the supporting structure whilst not compromising the lattice structure.

Progressing on from this simple method of support, an algorithm was generated that operated on geometry rather than the slice data. With further processing after creating the lattice data the elements that share a particular point can be identified. With this information it can be determined whether a point is adequately supported by a previous link that is at least a particular angle from the horizontal. The process for this is detailed in Figure 95.

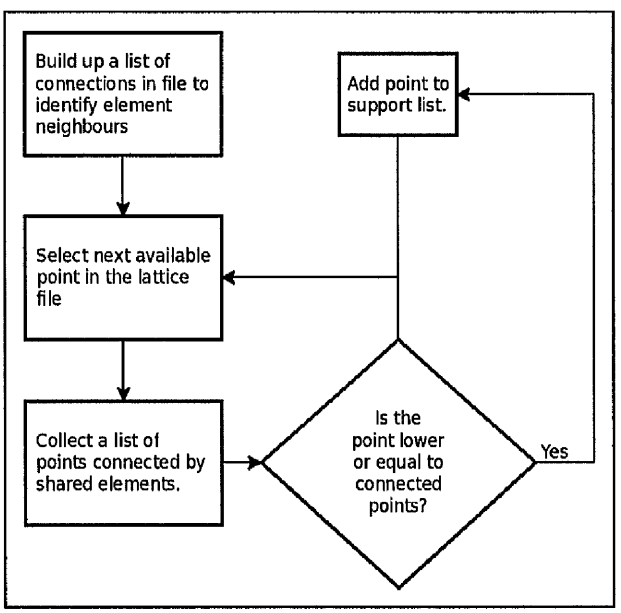


Figure 95: Identifying points that require supporting.

Figure 96 shows the shaped lattice structure previously detailed in Figure 89. Here the shaped lattice structure has been processed to create a set of supporting pillars.

This processing would benefit from further optimisation to recognise when a support pillar goes through a lattice region, and cross linking of the support pillars to improve the supporting structures tolerance to errors during build.

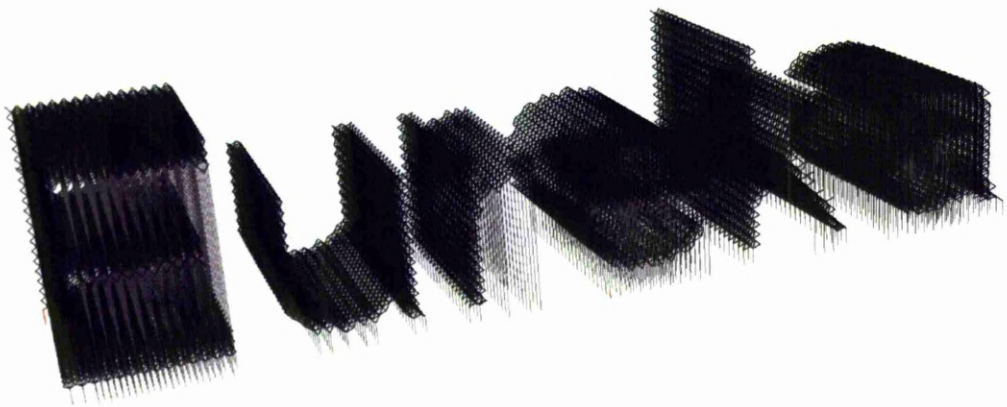


Figure 96: Supporting a lattice geometry created as a visualisation for a magazine article.

6.1.5.5 Importing and Exporting of lattice Geometries

The use of the 3D lattice geometry over the direct to slice method gave another fundamental benefit. The software was set up so that the lattice structures could be output after creation or imported from external sources into the lattice generation and slicing software. The output option allowed users to save a lattice geometry before it was output to slice data. This could save time regenerating the lattice in the event of a computer/power failure, or allow output into FEA formats to further the simulation modelling of the structures. The input option allowed other researchers the ability to create 3D geometries that could then be imported and sliced, and formatted ready to be built on the Realizer machine. This also allowed the users access to the support generation functions developed for the 3D structures. One such structure imported from an external source is a lattice structure created to be conformal to a pair of surfaces. The illustration of this

structure provided by the researcher is pictured in Figure 97.

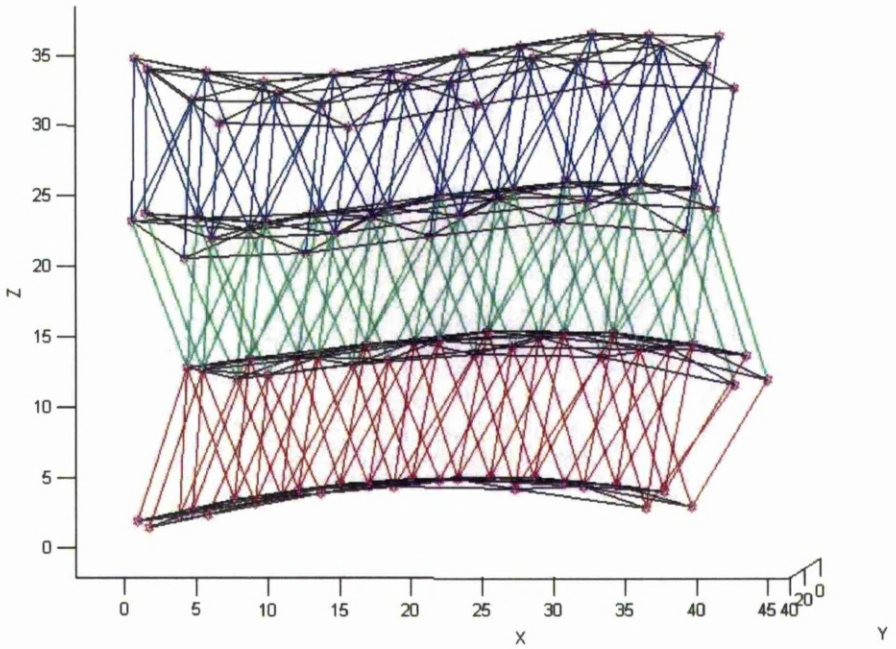


Figure 97: Lattice element model created by an external collaborator.

Once imported this lattice structure was supported and visualised in the software as shown in Figure 98.

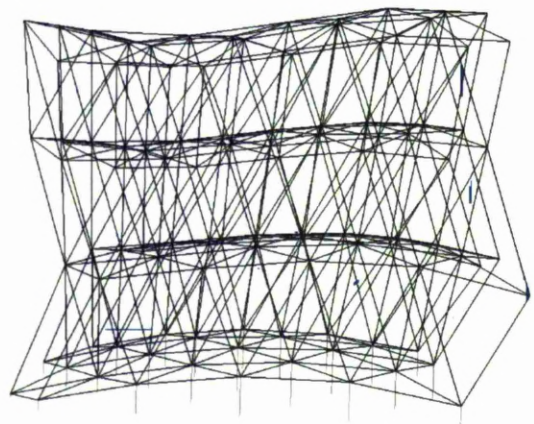


Figure 98: Visualisation of lattice created by external collaborator imported and supported.

6.1.6 Shaping the Slice Data

6.1.6.1 2D Membership Tests

Over the course of the research the method used to shape the computer model of the lattice structure before it was built progressed through a number of different iterations. The closer the clipping occurred to the output of the data to the slice file the simpler the concept, but in practice the least usable due to the large datasets created at each step.

The first iteration of shaping the data was by creating the lattice structure and outputting the slice points direct to a slice file without referencing any CAD geometry. A set of points for one slice plane would be collected then all exported into a slice file in a format that could be used by the SLM machine. This method was limited to regular blocks of lattice whose cell size was divisible into the extents of the lattice an integer number of times.

The simplest method to create a lattice structure shaped to CAD data was to

generate a block of lattice geometry to the same, or slightly larger extents as a user specified STL CAD file and slice it. Following this the supplied CAD file was sliced at the same z-heights, this is shown in Figure 99. For clarity the lattice is sliced along side the CAD as an element model, whereas in reality if visualised they would occupy the same space.

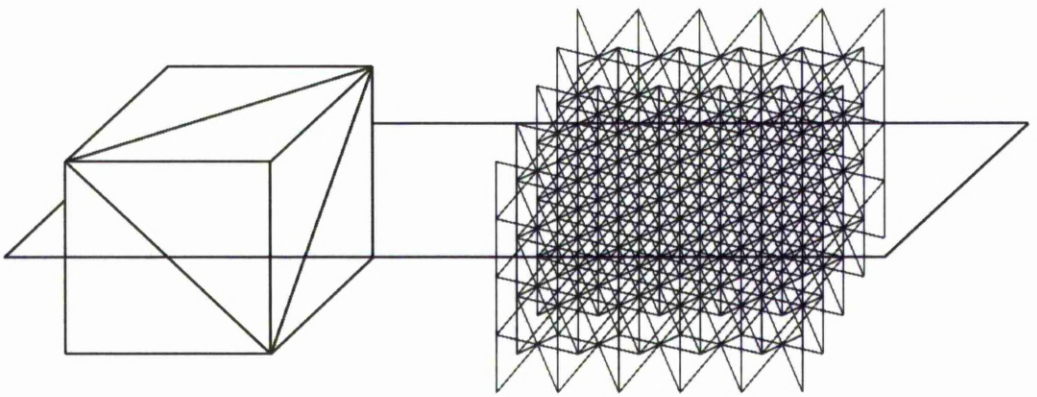


Figure 99: A simple cube STL file and lattice structure being sliced at the same time.

Each of these points were then checked to see whether they fell inside or outside of the contour. For simplicity the STL file was assumed valid and describing a closed, rather than infinite volume. This is a fair assumption as an open STL file is not valid, and errors would be apparent, and where possible could be fixed in other packages developed for layer manufacturing technologies.

The points membership to the CAD data was determined using a ray cast method. A line parallel to the y-axis (constant x value) was past from the maximum y-axis extent of the CAD data to the minimum y-axis extent as seen in Figure 100. The ray was then checked for intersections between the 2D contour and itself.

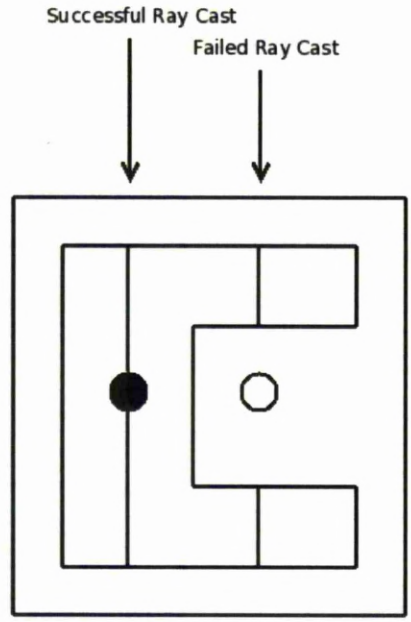


Figure 100: Two 2D ray cast demonstrations detailing a successful and unsuccessful membership test.

This was achieved by taking each vector segment of the contour in turn and;

- it's gradient and y-axis intersection calculated,
- the intersection between the ray and the infinitely long line $y = mx + c$ calculated,
- the intersection checked to see whether it is within the contour segment.

Once all the vectors have been checked the intersections are sorted in the order they occur along the direction of the cast ray. The first intersection indicates the beginning of an inner region on the CAD data and the following marks an exit from the inner region of CAD. As the points are counted along the length of the ray from the first each odd intersection indicated the following ray section was internal to the CAD and the subsequent even intersection marked the start of a section of ray that is external to the CAD data.

Once the sections of the ray which are inside the CAD data had been determined the initial point was checked against the ray sections to see if it fell within one of the sections inside the CAD. If so it belongs to the CAD data and is included in the final slice data. Figure 101 shows a single slice of lattice-slice intersection points where the hollow points have been rejected due to being determined as outside the contour by the ray cast check.

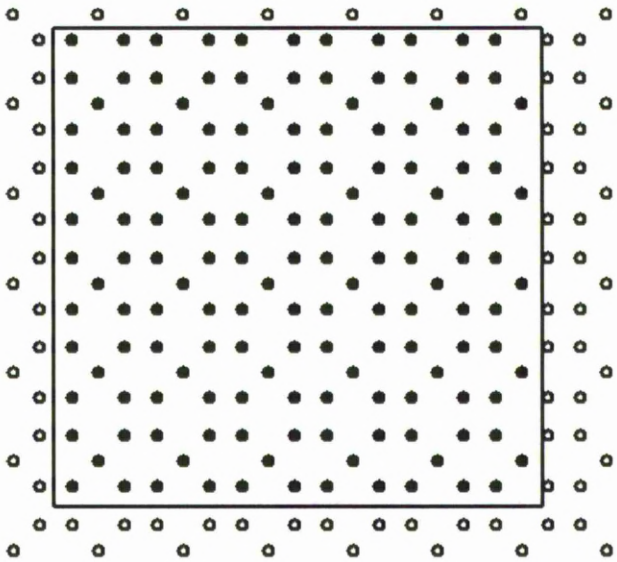


Figure 101: The sliced data from the lattice overlaid on the slice contour from the simple cube detailed in figure 99. All points have been subjected to a membership test. Points inside are shown filled.

Contour segments whose start and endpoint have the same x axis value are considered to have no intersection with the cast ray. This is because to be valid the STL file needs to describe a non-zero closed volume. If this is the case then even if a ray perfectly aligns with a vector segment of a contour then it will intersect with a vector after, and before – not necessarily the vector immediately before or after though as it is possible to get two or more aligned vector segments.

An intersection with a start of one vector and the end of the adjoining vector was considered as one intersection rather than two. If it was considered as two separate

intersections this would invert the ray after the double intersection.

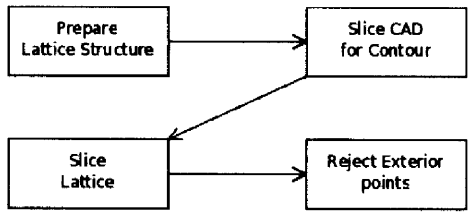


Figure 102: Flow chart detailing the steps that have taken place to generate the information detailed in figure 101.

This method of shaping the lattice structure simplified in Figure 31 worked for the iteration of the software that created randomised structures too. In this iteration of the software the whole lattice structure was created in advance so the location of the lattice element joints could be randomised without risking creating a discontinuous structure that would be unlikely to build correctly.

6.1.6.2 3D Lattice Clipping

While the software was only creating completely regular structures the intersection points were calculated based on the slices location in the lattice structure. The following iteration clipped a 3D lattice structure to fit within the STL file. Slicing of the lattice at a particular z-height resulted in collection of points – all of which where inside the sample CAD data. The process required to generate the slice data is simplified to that seen in Figure 103, but the actual process of clipping the lattice structure relies on more complicated geometric operations which are discussed later.

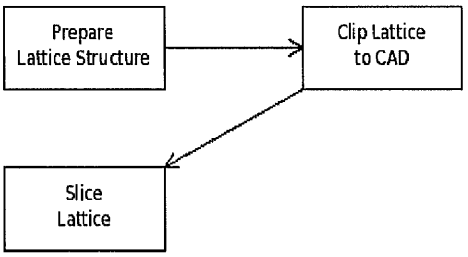


Figure 103: Flow chart detailing an alternative process to that shown in figure 102. Here the 3D lattice structure is being clipped to the shape of the lattice structure.

As the complexity of the lattice structures increased, and the proportion of the volume of the maximum extents cuboid of the CAD data to that contained by the CAD data dropped this method became less efficient and the lattice elements began to be clipped to the shape for the CAD data. This resulted in all of the intersection points between the lattice structure and the slice plane being passed through to the slice file without the need for membership tests on each lattice intersection point.

The clipping of the lattice element to the CAD data is far more complex than membership of a point to the area contained by a contour. The CAD file is a continuous surface described by a cloud of points linked by a triangular mesh. So the only information that can be directly extracted from the CAD data – with limited post processing - is points, triangles, triangle edges, and with a little more post processing normal, triangle, and edge neighbour information.

To clip the lattice the concept of ray casting has to be expanded to three-dimensions. Two functions were developed, membership check for a single point, and clipping of the element to supplied CAD data. Figure 104 shows a z-ray being used as part of the process to check for a point's membership to the volume of CAD data.

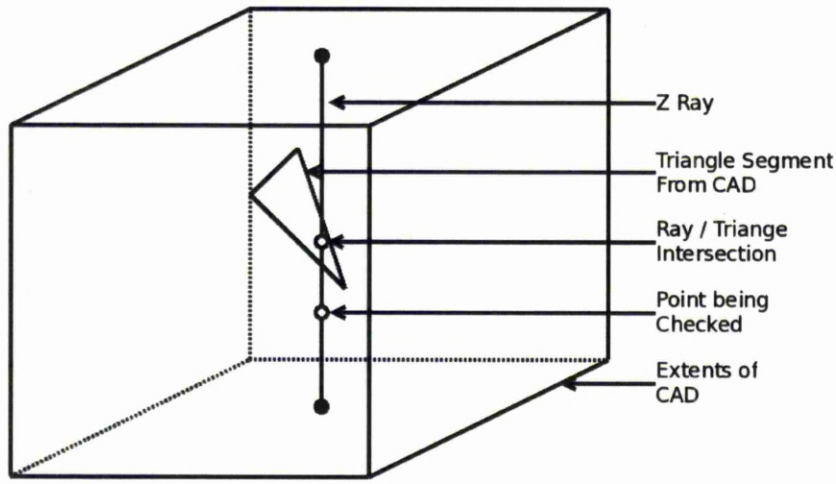


Figure 104: Z-Ray being compared to a single triangle segment from CAD data in order to check for a point's membership to the volume of CAD area.

The point/CAD membership test utilised a ray cast through the maximum z axis extents of the CAD at the points x and y axis coordinate values. The following procedure is used to find intersection points between the ray and the CAD on each triangle:

- Calculate the extents of the triangle.
- Disregard any triangles that describe no area when projected parallel to the z axis onto the x/y plane. (if no area, then the triangle is parallel to the element and so there is no single intersection).
- Determine whether the ray passes through the extents points (if it doesn't then there is no intersection).
- Ignoring the z-axis component check to ensure the x/y coordinate falls within the triangle using the previously developed 2D contour membership test. (if the x/y coordinate is outside the 2D decomposition of the triangle

then there is no intersection)

- Calculate the z-axis component of the intersections between the ray cast and the x/y triangle contour. An illustration of this is shown in Figure 105.
- Finally based on the two intersection points above calculate the z-axis component of the intersection between the discrete triangle plane and the z-axis parallel ray.

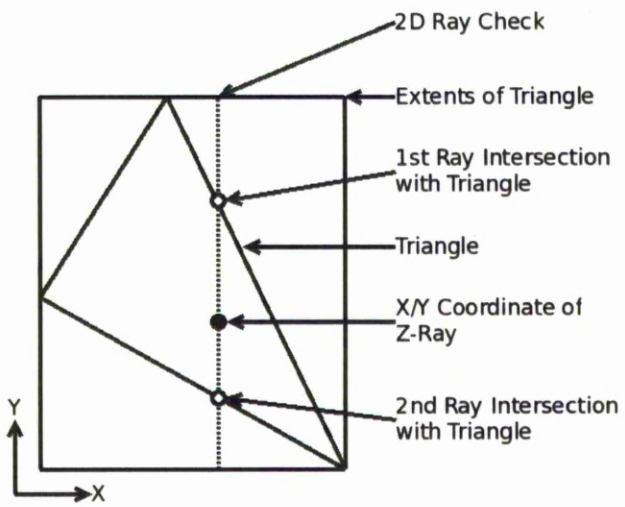


Figure 105: Using a 2D projection of the z-ray triangle intersect to find the z-component of the intersection point.

After all of the intersection points have been found for the ray they are ordered as in the same way as the 2D ray cast and the internal sections of the ray are identified. If the point falls on one of these points then it is internal to the CAD.

The clipping of the lattice elements so that they filled the CAD required the ray casting be developed into a full 3D check, rather than an axis parallel check – which in essence is only a slight extension to the 2D ray check.

To clip the elements each triangle in the CAD file needed to be checked for intersection between it and the infinitely long line on which the start and end

points of the element belong. This intersection point was subsequently checked to see if if fell between the points of the element on the line.

- Find the equation for the infinite line described by the element, then for each triangle:
    - Find the equation of the infinite plane described by the three points of the triangle.
    - Ensure the line is not parallel to the triangle (if so, no intersection is recorded).
    - Calculate the intersection between the line and the plane.
    - Check to see whether this point is in the discrete area described on the infinite plane by the triangles three points.
- Collect all intersections, sort and perform a membership test on the centre of each subsection of the line to determine whether that part of the ray is internal to the CAD.
- Perform a boolean operation on the element overlaid on the sections of the ray internal to the CAD and reject sections of the element outside these sections.
- Reject any sections of the element not connected to the original start or end of the element. The rejected sections will be unconnected to the rest of the lattice and so unable to contribute to the mechanical properties of the lattice or build correctly.

6.1.6.3 Element – Triangle 3D Intersect

Finding an intersection between a triangle and lattice element requires multiple 3D vector geometry calculations. The basis of these steps is outlined in the following section. At the start of the calculations an assumption is made that there is a finite point intersection between the element and the triangle. This situation is illustrated in Figure 106.

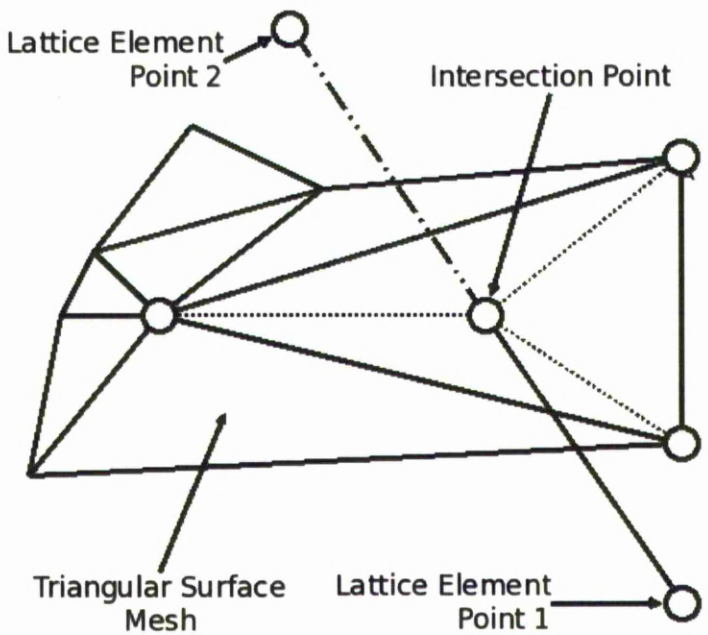


Figure 106: An intersection between a lattice element and a triangle within a mesh.

Figure 107 identifies the vertexes of the triangle and element being checked for intersection. In this figure there is no intersection with the element, however there is an intersection with the infinitely long line that the element's end points belong to. The types of intersections that could occur between the element and the triangle are;

- no intersect,
- point intersect,

- and the line being co-existant with the triangle.

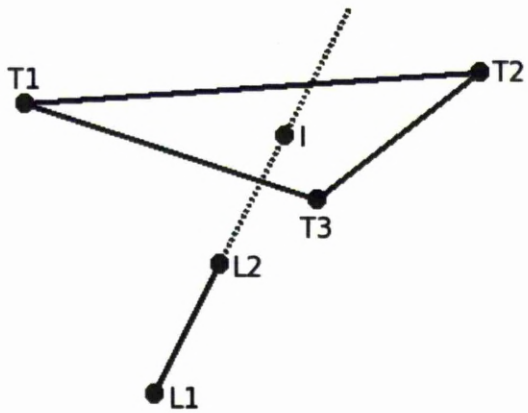


Figure 107: A projected intersection of an element through a plane.

The components of the vertexes identified in Figure 107 are detailed in Figure 108. T1 through T3 are the vertex of the triangle. L1 and L2 are the end points of the lattice element. I is the intersection between the infinitely long line that the lattice element is a section of and the triangle.

$$T_1 = (x_{t1}, y_{t1}, z_{t1})$$
$$T_2 = (x_{t2}, y_{t2}, z_{t2})$$
$$T_3 = (x_{t3}, y_{t3}, z_{t3})$$

$$L_1 = (x_{l1}, y_{l1}, z_{l1})$$
$$L_2 = (x_{l2}, y_{l2}, z_{l2})$$

$$I = (x_{i2}, y_{i2}, z_{i2})$$

Figure 108: Components of triangle, element, and intersection points.

The three vertexes of the triangle describe a finite region on an infinite plane. This is detailed in Figure 109.

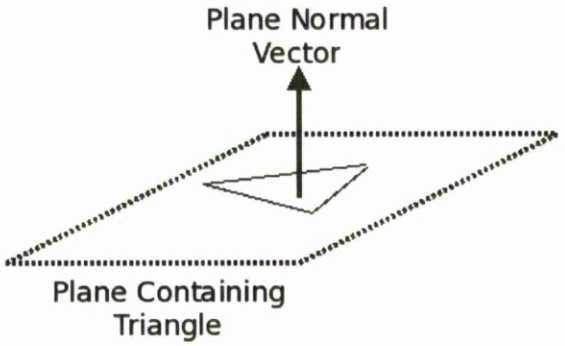


Figure 109: Triangle's parent plane and normal.

The equation for a 3D plane is given in Figure 110. The A, B, C components of the equation give x, y, and z components of the normal of the plane as shown in Figure 109. The D component of the equation gives the distance from the origin (0,0,0) in multiples of the normal vector given by the A, B, and C components. If the magnitude of the normal vector is 1, then D is the shortest distance from the plane to the origin.

$$Ax + By + Cz = D$$

Figure 110: Equation for a three dimensional plane.

In order to find the equation of the triangle two vectors are required that exist on the triangle, have non-zero magnitude, and are not co-existent. If the STL format triangular mesh file is valid that supplied the triangle, then the triangle will contain a non-zero area on the plane. In order for this to be true none of the points are co-existent. Due to this the vectors described by moving from point 1 to point 2, and point 1 to point 3 of the triangle can be used. The vector described by moving from point 1 to 2 of the lattice element is also required later. These three vectors are described in Figure 111.

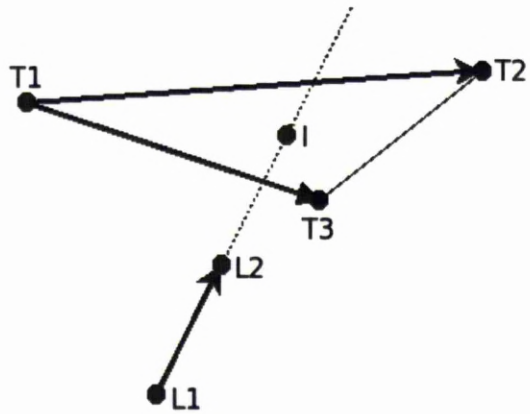


Figure 111: Vectors used to describe the triangles parent plane and the elements parent line.

The calculation of the vectors from the components detailed in Figure 111 and the names given to the components of the vectors is given in Figure 112.

$$V_{t1} = T_1 \rightarrow T_2 = \begin{bmatrix} x_{t2} \\ y_{t2} \\ z_{t2} \end{bmatrix} - \begin{bmatrix} x_{t1} \\ y_{t1} \\ z_{t1} \end{bmatrix} = \begin{bmatrix} x_{vt1} \\ y_{vt1} \\ z_{vt1} \end{bmatrix}$$

$$V_{t2} = T_1 \rightarrow T_3 = \begin{bmatrix} x_{t3} \\ y_{t3} \\ z_{t3} \end{bmatrix} - \begin{bmatrix} x_{t1} \\ y_{t1} \\ z_{t1} \end{bmatrix} = \begin{bmatrix} x_{vt2} \\ y_{vt2} \\ z_{vt2} \end{bmatrix}$$

$$V_l = L_1 \rightarrow L_2 = \begin{bmatrix} x_{l2} \\ y_{l2} \\ z_{l2} \end{bmatrix} - \begin{bmatrix} x_{l1} \\ y_{l1} \\ z_{l1} \end{bmatrix} = \begin{bmatrix} x_{vl} \\ y_{vl} \\ z_{vl} \end{bmatrix}$$

Figure 112: Calculating the vectors for the parent plane and line.

The cross product of the two vectors from the triangle gives the components of the normal vector of the plane. The calculation of the cross product from the components detailed in Figure 108 is given in Figure 113.

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} (y_{vt1} \times z_{vt2}) - (z_{vt1} \times y_{vt2}) \\ (z_{vt1} \times x_{vt2}) - (x_{vt1} \times z_{vt2}) \\ (x_{vt1} \times y_{vt2}) - (y_{vt1} \times x_{vt2}) \end{bmatrix}$$

Figure 113: Taking the cross product of the triangle vectors to find the normal.

The normal vector given by the equation in Figure 113 doesn't necessarily have a magnitude of 1. In order to modify the vector to ensure it has a magnitude of 1 each component has to be divided by the magnitude of the vector. This is referred to as normalising the vector and is shown in Figure 114.

$$\begin{bmatrix} A_n \\ B_n \\ C_n \end{bmatrix} = \frac{1}{\sqrt{A^2 + B^2 + C^2}} \times \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Figure 114: Dividing the normal vector by it's magnitude to find the normal unit vector.

Once the vector has been normalised the equation of the plane can be modified to give the distance to the origin. This is shown in Figure 115. The A, B, and C components of the equation of a plane describe a family of parallel planes, the D component describes a distinct plane from the family. The equation can be re-arranged and a point from the triangle substituted in to find the value of the D component.

$$Ax + By + Cz + D = 0$$

$$D = -(Ax_{t1} + By_{t1} + Cz_{t1})$$

Figure 115: Substituting the first point of the triangle into the equation for a plane to find D.

When the components A, B, and C give a normal vector with a magnitude of 1 the

equation of the plane through the triangle can be re-arranged to give the shortest distance from any arbitrary point to the plane as shown in Figure 116. A second plane is created that intersects the arbitrary point and is parallel to the first. The equation of the plane is re-arranged to resolve the D component for the new plane. The distance between the planes is the absolute value of the D component of the new plane subtracted from the D component of the first plane. The equation used to resolve the D component is substituted in place of the second D value to give a final equation. The distance between the planes is the shortest distance from the plane to the point.

$$P_p = (x_p, y_p, z_p)$$

$$\begin{gathered} Ax + By + Cz + D = 0 \\ Ax_p + By_p + Cz_p + D_p = 0 \\ Ax_p + By_p + Cz_p = -D_p \end{gathered}$$

$$\sqrt{A^2 + B^2 + C^2} = 1$$

$$\begin{gathered} d = \text{shortest distance}(\, P_p \rightarrow Plane \,) \\ d = |D - D_p| \end{gathered}$$

$$d = |D + Ax_p + By_p + Cz_p|$$

Figure 116: Detailing the use of the equation of a plane to determine the shortest distance to an arbitrary point.

The equation in Figure 116 can be used to find the shortest distance from the first point of the lattice element to the plane and the second point of the lattice element to the plane. This is detailed in Figure 117. The ratio of these can be arranged to give the multiple of the vector of the element that needs to be added to the first point of the element to reach the intersection point.

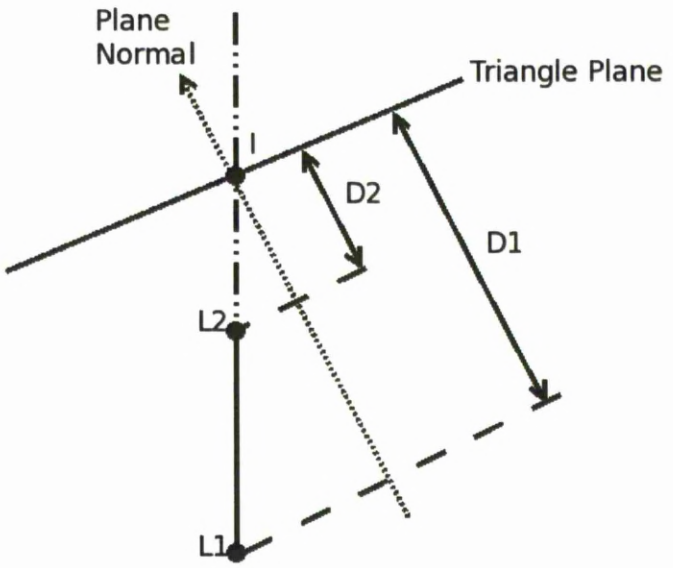


Figure 117: Detailing the shortest distance from the points of the element to the triangles parent plane.

The equations in Figure 118 show the use of the formulas in Figure 116 to find the distances labelled in Figure 117. As well as showing the equations to find the distances from the lattice element points to the plane it also shows that the distance from the intersection point to the plane is 0.

$$d_{l1} = Ax_{l1} + By_{l1} + Cz_{l1} + D$$
$$d_{l2} = Ax_{l2} + By_{l2} + Cz_{l2} + D$$
$$0 = Ax_{i} + By_{i} + Cz_{i} + D$$

Figure 118: Calculating the distance from the points of the element and the intersection point to the plane.

Figure 119 Shows the addition of a multiple of the vector of the element detailed in Figure 111 and Figure 117 to the first point of the lattice element to reach the intersection point.

$$\begin{bmatrix} x_{l1} \\ y_{l2} \\ z_{l3} \end{bmatrix} + \mu \begin{bmatrix} x_{vl} \\ y_{vl} \\ z_{vl} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Figure 119: Equation describing the intersection with the plane as a multiple of the vector from the element's first point to its second summed to the element's first point.

Figure 120 uses the equation showing that the distance from the first point of the lattice to the plane is the same as a multiple of the distance from the first to the second. This equation is re-arranged to give an equation for μ. Following this the equations from Figure 118 are substituted in to replace the distances from the points of the lattice to the plane. If the denominator for the final equation is zero then there is either infinite intersection, or none. This indicates the lattice element is parallel to the plane of the triangle.

$$d_{l1} - \mu\left(d_{l2} - d_{l1}\right) = 0$$

$$-\mu = \frac{d_1}{d_2 - d_1}$$

$$\mu = \frac{d_1}{d_1 - d_2}$$

$$d_{l1} = Ax_{l1} + By_{l1} + Cz_{l1} + D$$

$$d_{l2} = Ax_{l2} + By_{l2} + Cz_{l2} + D$$

$$\mu = \frac{Ax_{l1} + By_{l1} + Cz_{l1} + D}{\left(Ax_{l1} + By_{l1} + Cz_{l1} + D\right) - \left(Ax_{l2} + By_{l2} + Cz_{l2} + D\right)}$$

$$\mu = \frac{Ax_{l1} + By_{l1} + Cz_{l1} + D}{A\left(x_{l1} - x_{l2}\right) + B\left(y_{l1} - y_{l2}\right) + C\left(z_{l1} - z_{l2}\right)}$$

Figure 120: Calculations used to find the multiple described in Figure 117 using ratios of the shortest distances from each point on the element to the plane.

Once the value of the multiple is known the intersection point between the infinite line to which the lattice line belongs and the infinite plane that the triangle is part of can be calculated using the equation from Figure 119. It is simple to see if this point falls within the lattice line, but more complex calculations are required to determine whether an intersection point that falls within a lattice line also intersects the area described by the triangle as shown in Figure 120.

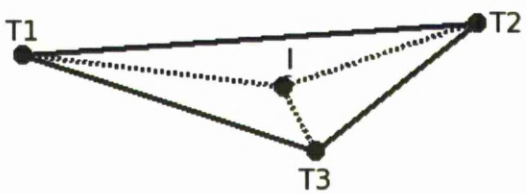


Figure 121: Detailing the location of the intersection point relative to the triangles vertexes.

To determine whether the intersection point falls within the triangle the sum of the angles between the vectors from the intersection point to the three vertex of the triangle need to be calculated. The components of the vectors are calculated in Figure 122.

$$V_{i1} = I \rightarrow T_1 = \begin{bmatrix} x_{t1} \\ y_{t1} \\ z_{t1} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{vi1} \\ y_{vi1} \\ z_{vi1} \end{bmatrix}$$

$$V_{i2} = I \rightarrow T_2 = \begin{bmatrix} x_{t2} \\ y_{t2} \\ z_{t2} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{vi2} \\ y_{vi2} \\ z_{vi2} \end{bmatrix}$$

$$V_{i3} = I \rightarrow T_3 = \begin{bmatrix} x_{t3} \\ y_{t3} \\ z_{t3} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{vi3} \\ y_{vi3} \\ z_{vi3} \end{bmatrix}$$

Figure 122: Calculating the three vectors from the intersection point to the triangle's vertexes.

The dot product of two vectors of magnitude 1 gives the cosine of the angle between them. The first step towards calculating these angles is therefore normalising the vectors as detailed in Figure 123. The process to normalise a vector was detailed previously in Figure 114.

$$V_{i1n} = \text{normalise}(V_{i1})$$
$$V_{i2n} = \text{normalise}(V_{i2})$$
$$V_{i3n} = \text{normalise}(V_{i3})$$

Figure 123: Finding the unit vectors of the intersection vectors.

Figure 124 labels the angles between the vectors for clarity in the following step.

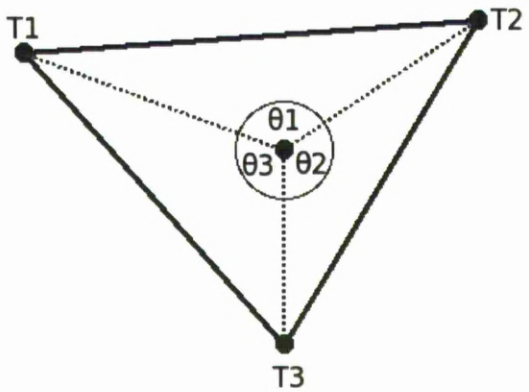


Figure 124: Detailing the angles between the intersection vectors.

The dot product is calculated to determine the cosines of the angles between the three intersection vectors detailed in Figure 121 and Figure 124. As shown in Figure 125 the intersection occurs within the triangle when the sum of the arc-cosines of theses cosines is equal to 2π.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = (x_1 \times x_2) + (y_1 \times y_2) + (z_1 \times z_2)$$

$$\begin{aligned} a_1 &= V_{i1n} \cdot V_{i2n} \\ a_2 &= V_{i2n} \cdot V_{i3n} \\ a_3 &= V_{i3n} \cdot V_{i1n} \end{aligned}$$

$$\begin{aligned} \theta_1 &= \arccos(a_1) \\ \theta_2 &= \arccos(a_2) \\ \theta_3 &= \arccos(a_3) \end{aligned}$$

$$2\pi = \theta_1 + \theta_2 + \theta_3$$

Figure 125: Calculating the sum of the angles between the intersection vectors using the dot product of the unit intersection vectors to prove the intersection point lies within the triangle's vertexes.

6.1.7 Optimising Use of Computing Resource

Creating a machine specific slice file for a lattice structure comprises of three distinct stages of computation which were performed in separate sections. The stages were:

1. The creation of the 3D lattice dataset.
2. Recursively slicing the structure at intervals equal to the layer height of the build.
3. Converting the data into the format specific to the machine and writing the resulting string to a file.

Through some basic investigations into the performance of string, list, and file handling this was found to be an inefficient way of processing the files. Simple programs recorded the amount of time taken to return the value of items at random positions in a list and string, the time taken to append a fixed number of characters to a lists and strings of increasing size, and the time taken to write files of increasing sizes in various numbers of write operations.

All of the tests for strings, lists and files where performed on an AMD TL-60 dual core laptop with 4GB of memory running Linux. While running the tests the network connection and all other programs where closed down so that the test stood the least chance of being disturbed. The sizes of the tests where chosen so that the memory usage during the tests did not exceed the available RAM on the machine, and so attempting to avoid the increase in memory access times associated with using page files on the hard disk, which in this case is the standard IDE PATA magnetic platter type, rather than the far quicker more recently

commercially available SATA, SATA II, or PCIE solid state drives.

6.1.7.1 String Timing Functions

A string is an object comprising of a collection of characters each of which are a byte with the value of 0-255. When bytes are read from a file the bytes are supplied as a string. When a number of bytes are wrote to a file these are passed in a string. In the direct to slice version of the lattice generation software the whole file is written to one string before the string is written to the file.

```
1. def TimeSeeks(self):
2.     totalTime = 0.0
3.     for i in xrange(self.numOfOperations):
4.         itemKey = random.randrange(self.numOfItems)
5.         startTime = time.time()
6.         item = self.container[itemKey]
7.         totalTime += time.time() - startTime
8.     return totalTime


1. def TimeAppending(self):
2.     totalTime = 0.0
3.     for i in xrange(self.numOfOperations):
4.         startTime = time.time()
5.         self.container += "a"
6.         totalTime += time.time() - startTime
7.     return totalTime
```

Figure 126: Timing functions for string seek, and append operations.

The first function in Figure 126 shows the code used to time a number of seek operations on a list, and the second shows the timing function used to record the time used to append a number of times onto an existing string. Before these functions are used the string object is created and named "self.container". "self.numOfOperations" is set to the total number of additions made to the string. "time.time()" is a call to a function which returns the system time expressed as a decimal value in seconds, finally "self.numOfItems" is the length of the string object.

When this function is called the lines 4-7 are repeated once for every required seek operation. "random.randrange(self.numOfItems)" returns a random number which give the location for the following seek opperation. This number returned from this function has an even distribution of probabilities, so any location on the string has an even probability of being selected. The seek operation takes place on line 6. The system time is recorded prior to the function on line 5, and the expired time is calculated and added to the running total on line 7. The time is recorded as close to the seek function as possible.

The second function "TimeAppending" shows a similar function which is used to test the time taken to add a number of characters to the string. The timed line in this function is "self.container += "a"". The += sign is an instruction to add the object to the right of the sign to the string to the left, using the name for the old string on the left for the result.

6.1.7.2 List Timing Functions

A list is an object which can hold an ordered set of objects, or more correctly references to objects. For example it can hold floats and integer values, strings, functions, of instances of classes. Unlike the string the object is mutable, which means things can be added or removed from the list without having to create a new list object.

```
1. def TimeSeeks(self):
2.     totalTime = 0.0
3.     for i in xrange(self.numOfOperations):
4.         itemKey = random.randrange(self.numOfItems)
5.         startTime = time.time()
6.         item = self.container[itemKey]
7.         totalTime += time.time() - startTime
8.     return totalTime


1. def TimeAppending(self):
2.     totalTime = 0.0
3.     for i in xrange(self.numOfOperations):
4.         startTime = time.time()
5.         self.container.append("a")
6.         totalTime += time.time() - startTime
7.     return totalTime
```

Figure 127: Timing functions for list seek, and append operations.

As can be seen in Figure 127 the functions testing the seek and append operations for the list are very similar to those for the string shown in Figure 126. The only difference is the use of “self.container.append("a")” to add the item to the list. “self.container += ["a"]” would have done the same thing, but a separate function is required to test the list as “self.container += "a"” would fail as it is trying to sum a list and a string object, rather than add a string object to a list. The “self.container.append("a")” method is the more common approach for adding items to a list.

6.1.7.3 File Timing Functions

```
 1. def TimeTest(self):
 2.     fileObj = open('test.txt', 'w')
 3.     timeExpired = 0.0
 4.     bytesPerChunk = self.numOfItems / self.numOfOperations
 5.     for i in xrange(self.numOfOperations):
 6.         start = i * bytesPerChunk
 7.         finish = start + bytesPerChunk
 8.         chunk = self.container[start:finish]
 9.         startTime = time.time()
10.         fileObj.write(chunk)
11.         timeExpired += time.time() - startTime
12.     os.remove('test.txt')
13.     return timeExpired
```

Figure 128: Timing function for file write operations.

Figure 128 Shows the code used to test the effect of writing files of different sizes, and in different number of write operations. During the set up of this test a string was created which was the length of the file to be written. This was stored in the object named "self.container". The line "self.container[start:finish]" extracts a slice from the start position of "start" to the finish position "finish". The file is written to through a file object. Before this can be used a file was opened, and this is done in line 2. The "self.numOfOperations" gives the number of writes that are given to complete the file, and "self.numOfItems" gives the number of bytes to be written to the file objects.

In this function lines 6-11 after the "for" line are repeated once for each section of the file write. The write to the file is completed on line 10, and once again the timing is done either side of this line. On completion of the test the created file is removed by the line "os.remove('test.txt')". This is done for convenience and is not timed.

6.1.7.4 Running the Tests – Bash Scripting

The programming language Python is a language which is not compiled as would be done for languages such as C, C++, and Java. In addition to this it is not purely interpreted from files with contents similar to that shown in Figures 127 and 128. When a Python file is first run it is converted by the interpreter into a byte-code format. While this is not compiling down to machine code it is making the code quicker to import as the file is reduced from the verbose format intended for human interaction into a binary format. In addition to this as a pure python library is used by a program it optimises itself. When trying to time the performance of a the fundamental functions such as those under investigation here that can be an issue, as the first test inside a python file may run notably slower than the following tests giving a false result that would be otherwise hard to interpret.

This was dealt with as follows in what is probably an over-cautious method, but intended to give the best indications of the underlying trends. The python script that runs the test either tests the list, or the string object, and tests either the time taken to run seek, or append tests. These are determined when the file is run on the command line by arguments passed to the script by the user, or bash script. The python file will have to test the string or list with a number of different size lists, or strings, over a number of different seek or append operations. The python script performs a full-factorial experiment over these values. However, due to the first run optimisation the full factorial experiment is re-run a number of times of that each test in the full factorial experiment happens at least once. Due to this level of repetition and the file sizes concerned the full experiment as run by the bash script in Figure 129.

```
 1. echo "Creating random character files."
 2. python JunkFileCreator.py
 3.
 4. fLimit=3
 5. oLimit=4
 6. testNum=0
 7. numTests=$((fLimit*oLimit*2*2))
 8.
 9. for contType in "list" "str"; do
10.     for testType in "seek" "append" ; do
11.         for ((fShift=0; fShift < fLimit ; fShift++)); do
12.             for ((oShift=0; oShift < oLimit ; oShift++)); do
13.                 echo "Running test " $((testNum+1)) " of "
$numTests; ((testNum++))
14.                 python RandomSeekTimer.py fileShift=$fShift
oppShift=$oShift containerType=$contType testType=$testType
15.             done
16.         done
17.         echo "Averaging Results."
18.         python SeekAppendResultsCompiler.py
containerType=$contType testType=$testType
19.     done
20. done
21.
22. testNum=0
23. for contType in "list" "str"; do
24.     for testType in "seek" "append" ; do
25.         for ((fShift=0; fShift < fLimit ; fShift++)); do
26.             for ((oShift=0; oShift < oLimit ; oShift++)); do
27.                 echo "Running test " $((testNum+1)) " of "
$numTests; ((testNum++))
28.                 python RandomSeekTimer.py fileShift=$fShift
oppShift=$oShift containerType=$contType testType=$testType
29.             done
30.         done
31.         echo "Averaging Results."
32.         python SeekAppendResultsCompiler.py
containerType=$contType testType=$testType
33.     done
34. done
35.
36. echo "Cleaning random character files"
37. python JunkFileCleaner.py
38.
39.
40. exit 0
```

Figure 129: Bash script used to run the seek and append tests for lists and strings.

A bash script, or shell script is a file that can be run on the terminal on a Linux machine which can be used to automate a sequence of terminal commands. The 40 line example given is used to run lines 14 and 28 over and over where

"$fShift" by the values 0, 1, 2, "$oShift" by 0, 1, 2, 3, "$contType" is replaced by "list" and "str", and finally "$testType" by "seek" "append". In the example given the python script would be run 48 times, and each run of the python script would be running 12 tests. Due to the high number of tests the bash script could take up to 12hrs to complete. The results however were collated, averaged, and formatted in comma delimited cvs format ready for import into a spread sheet program by an additional python script. Where error bars are presented they are showing the average error from the result set's average.

6.1.7.5 String & List Handling Results

Figure 130 shows the recorded time for 50,000 seek operations on a string varying from one to six million characters in length.

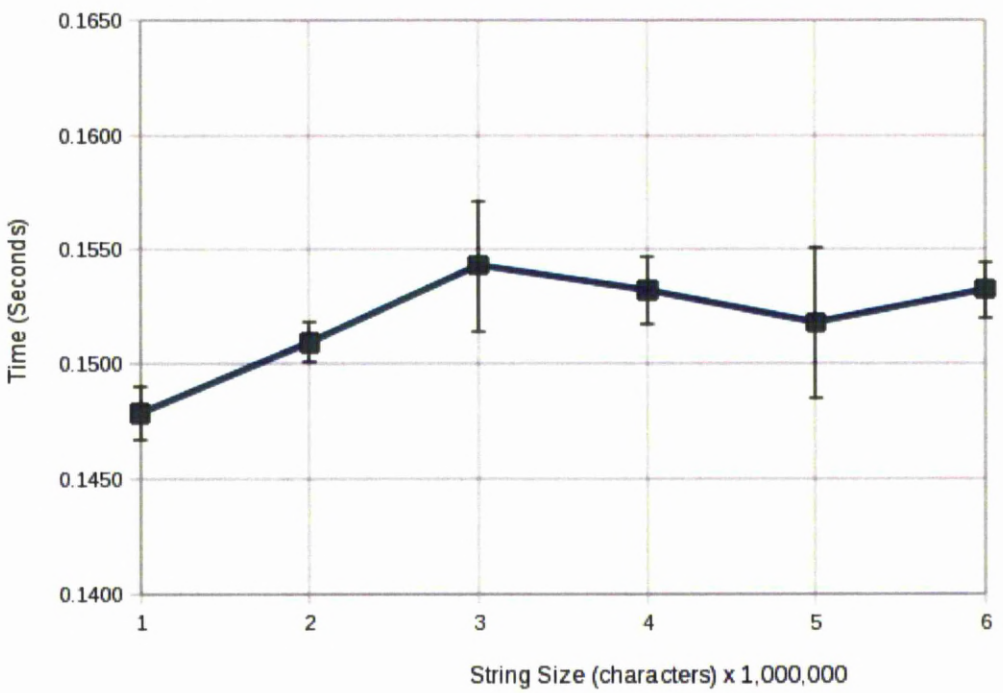


Figure 130: Time for 50000 seek operations on a string.

Figure 131 shows the recorded time for 50,000 seek operations on a list varying from one to six million characters in length.

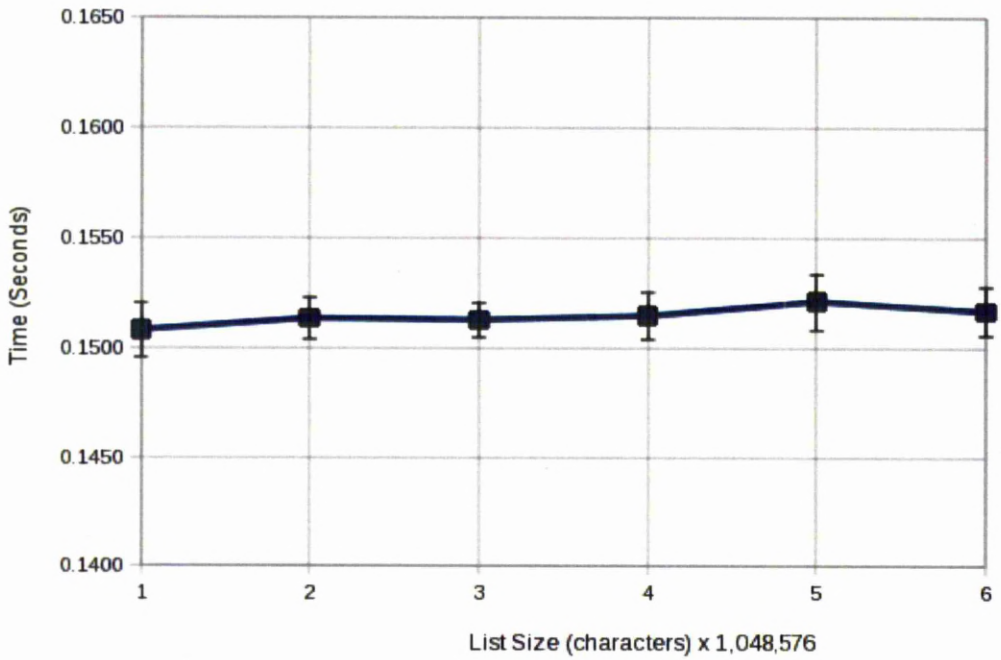


Figure 131: Time for 50000 seek operations on a list.

A string is an object containing a number of bytes and so is raw data, where as a list is a collection of objects. The list object generally out performs the string object, and the seek time for the list remains more constant with less variation. It is expected that the seek operation is having to search through the string, where as the list object has been better optimised for this and jumps directly to a specific address, directed by the key used.

Figure 132 shows a near directly proportional relationship (for the first five tests) of the time taken to add 50k characters, to the size of the original string. This is believed to be due to the string being an immutable. A mutable object such as the list can be changed, whereas the += operator on the string actually creates a new

string object by the addition of the string to its right and left, rather than adding the contents of the string to the right to the string on the left. So in the test case above the time is not for the addition of 50k characters, but the creation of a string of length 50k + the size of the original string.

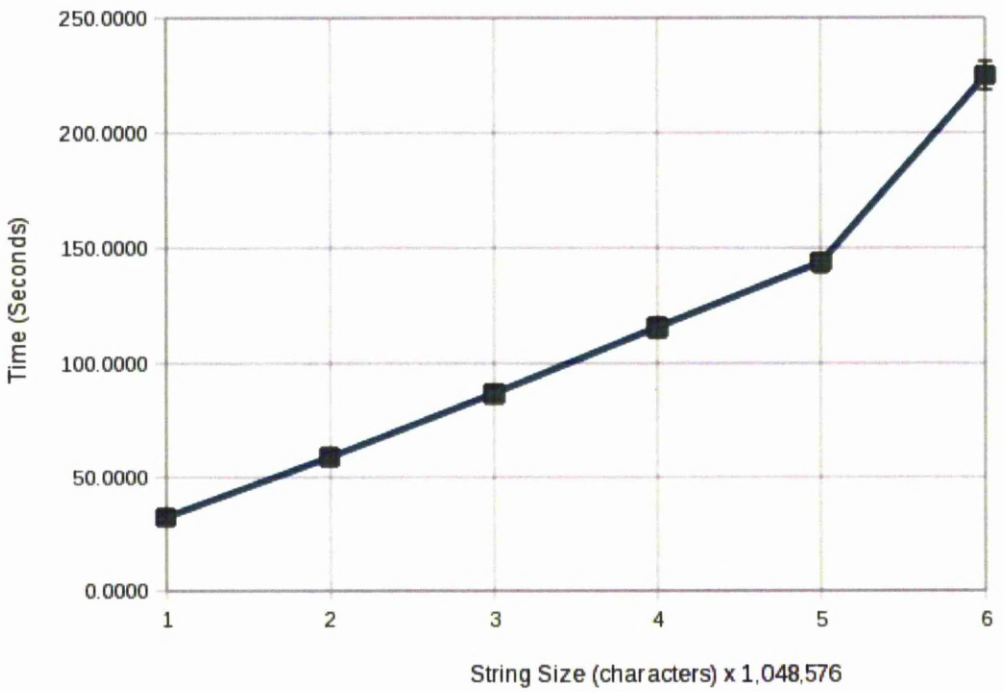


Figure 132: Time taken to add 50000 characters to a string.

Figure 133 demonstrates that appending items to a list object does not suffer the same problem as using the += operator with the strings. In this case the original list is a mutable object, and so the additional strings are added to the list, rather than creating a new list for each append operation.

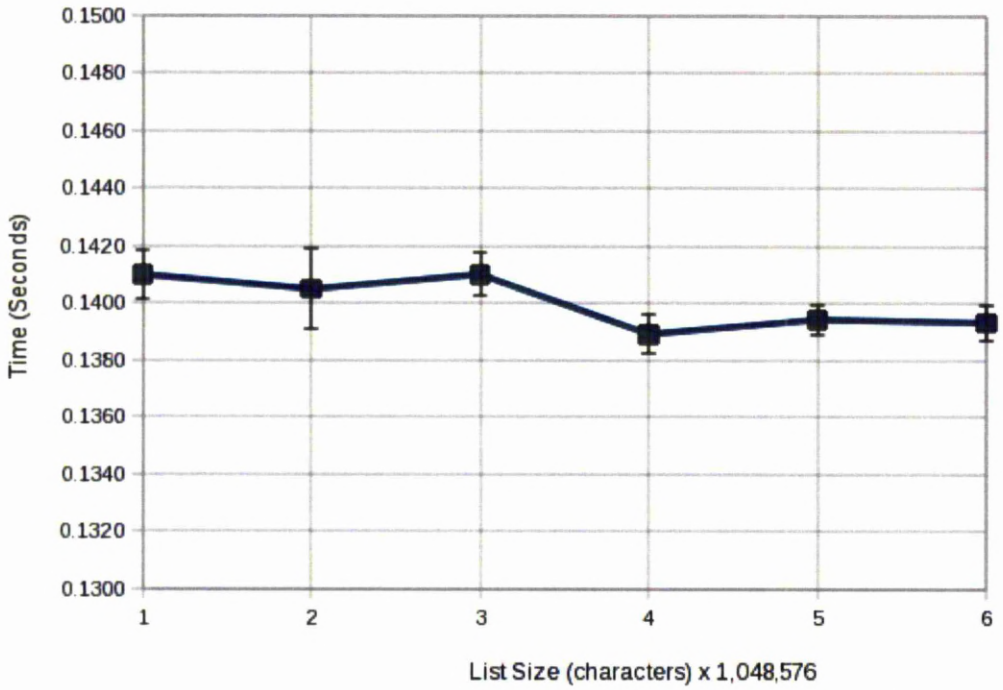


Figure 133: Time taken to add 50000 single character strings to a list.

6.1.7.6 File Handling Results

The file write times for files between the sizes of 1 and 600 megabytes shown in Figure 134 appear to demonstrated a sharp rise in processing time after 300 megabytes. In order to evaluate what is happening further tests where run focusing on this region.

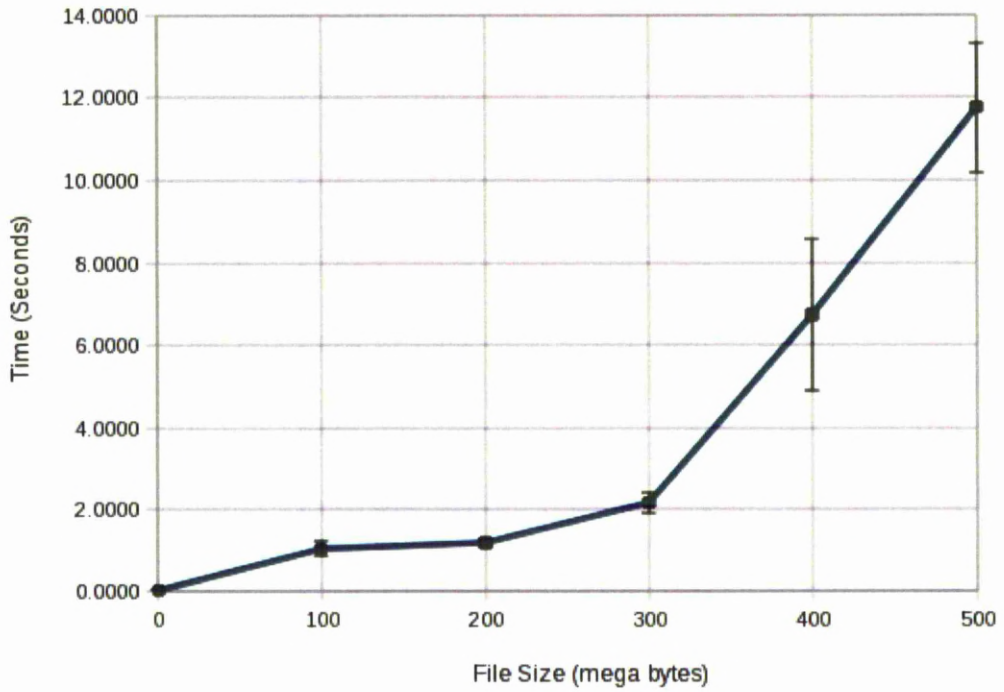


Figure 134: Write time for files between 1 and 500MB

Re-running the tests with more samples in the step region of the results as shown in Figure 135 show a close correlation to the original results before the sharp increase in file write times. During the more rapid increase in file write times after 300MB the results do not closely correlate with each other – regardless of the same test being run. Due to this the later tests are run in the range with good correlation. This could have been due to the size of the data file created by the software on the RAM having to be split up, rather than being able to be all in one area. The fragmentation of the data would be dependent on what else is in the RAM at the time of the test and how the operating system handles the software's request for extra resource. This could explain the increase in variation of the results during this period too.

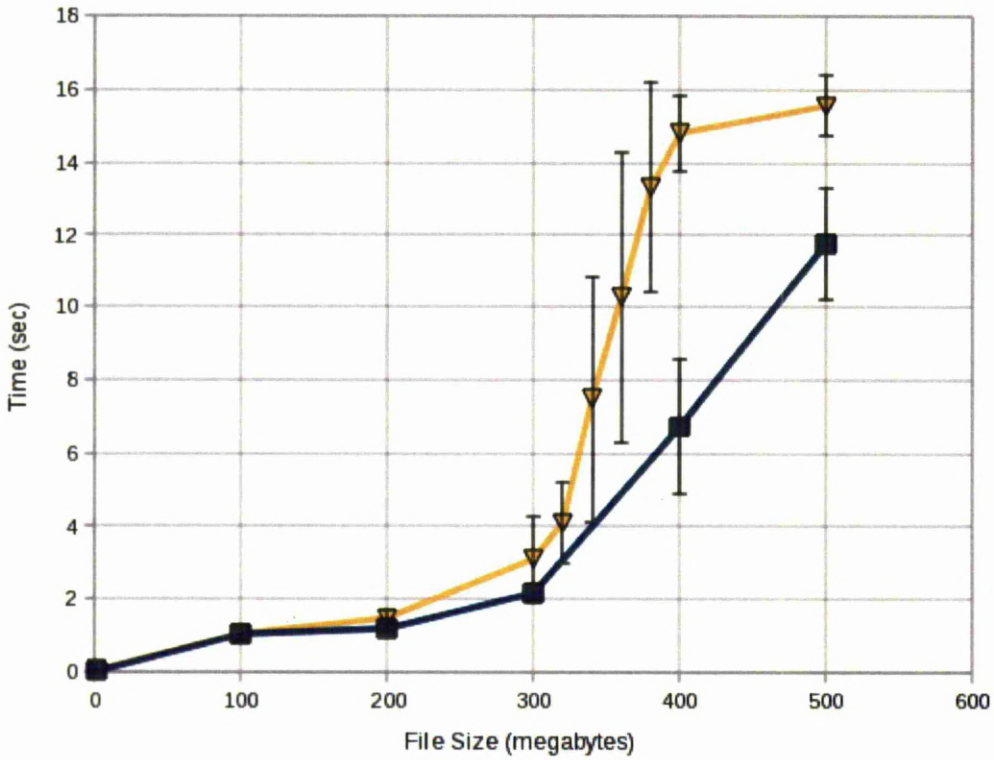


Figure 135: Stepped region results overlaid on original file write results.

In order to test the effects of splitting the writing of a file into multiple chunks the test focused on the 1 – 300MB file range and increased the resolution of investigation into the number of chunks to five tests ranging from 1 to 1000. The tests did show that this region of the tests were not linear, but also showed that there was little effect in file write time from splitting the file write into many stages.

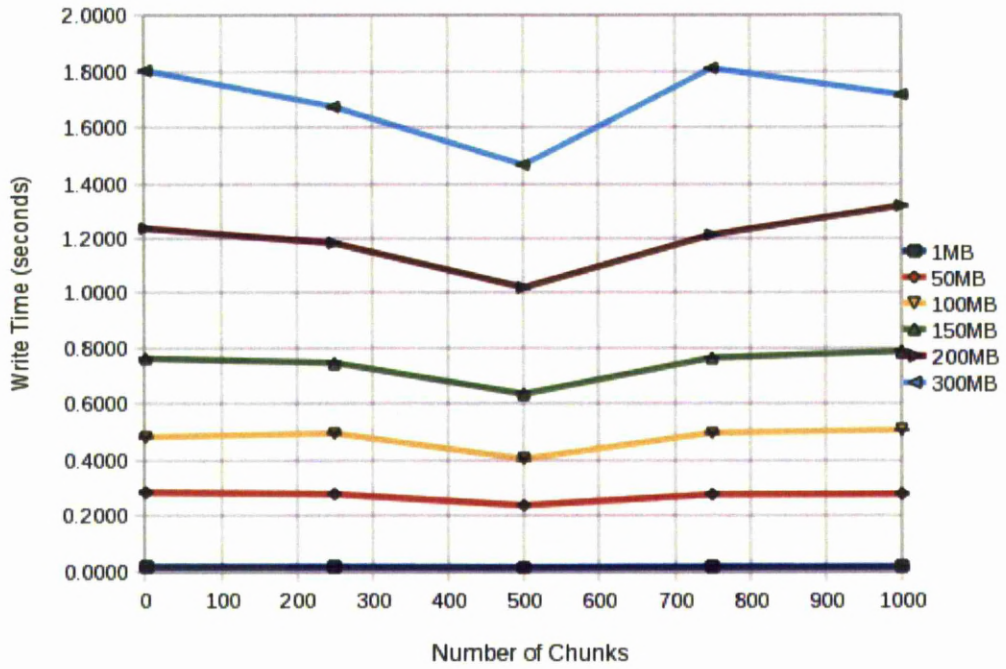


Figure 136: Write time for a file split in 1 to 1000 chunks.

On initial investigation it was assumed that there was a buffer which was being used to write the files to disk. To test this the file objects functions where studied and in addition to the open, close, read and write functions there is a flush function. This forces the writing of characters stored in file buffers into the files themselves. Figure 137 tested the effect of this function on writing a file in 4000 chunks. The effect of flushing the file did not seem significant when compared to the variance of the results, if anything it slightly slowed the write time.

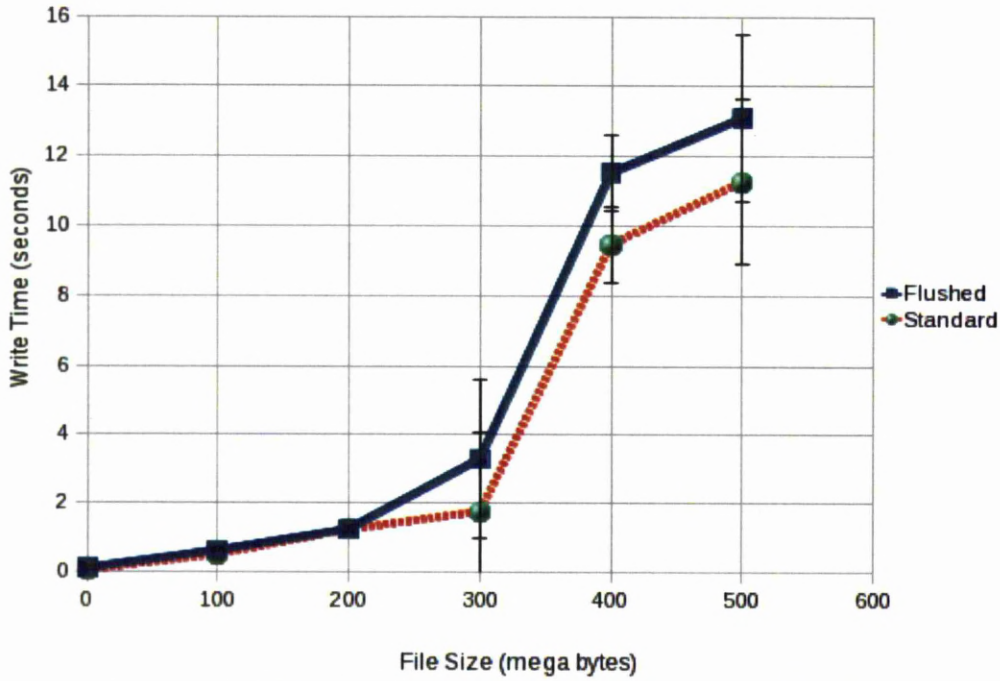


Figure 137: Write times for 4000 divisions, with and without file flush.

6.1.7.7 Timing Results Based Software Improvement

The slice data generation software with the staged processing architecture has been previously discussed. This was an evolution of the testing programs that were written in distinct sections. This allowed easier fault diagnosis during the development process and enabled the software development to be tackled in sections. The whole lattice could be created and viewed, clipped to the shape of the bounding CAD, then the whole slice data set created and viewed, and finally the slice data output file created. Creating the software in sections allowed initial testing on the machine to progress, but the results above show how this was not the most efficient way of writing the software.

The results of this testing showed that the original method of writing the whole

contents of the file to the string before writing to the hard disk was a inefficient approach. The initial logic was that writing the string before writing to disk would avoid disc seek times, but the results showed that splitting the file into may small writes does not have a significant effect on the time taken to write the file, at least between the range of one to one thousand writes for the file. In addition to this storing information to a string and extending the string is shown to be very inefficient as the software is effectively creating a new string each time due to the immutable nature of the string object. It shows that writing directly to disc rather than using a string object as a temporary holder is far more time efficient.

In addition to this there was no need to slice the whole file, then write the whole formatted file to disk. In order to reduce the required memory the software was reorganised to find all the intersections for one z-level, then create the single layer of slice data, then output this to the disk. This allowed the creation of larger and more complex lattice structures as unlike the previous method this did not require significantly more memory. This was a particular concern for the software running on 32bit Windows machines where there still is limit of about 1GB per process before a system level error is raised which crashes the running program by denying it's request for more memory space.

6.1.8 Pre Processing of the CAD data.

In order to create highly complex structures it was necessary to pre-process the CAD data. Without this all of the links in the lattice structure would need to be checked against all of the triangles in the file. If a high resolution, large CAD clipping surface was used or if the cell size was an order of magnitude smaller than the extents of the CAD then this would lead to significantly increased

processor time to create the lattices.

The processing time would be reduced if triangles in the vicinity of a ray check, or regions completely inside the STL file could be identified by other means. With this information any elements, or portions of an element that were within the region need not be checked and the number of triangles that have to be checked for lattice/surface intersections would be reduced. This information is not available in the native STL file format so further processing is required.

The example in Figure 138 shows a wire frame image of a simple arbitrary STL file with 20 triangles which will be used to demonstrate the process used to sub divide an STL file into cuboid regions.

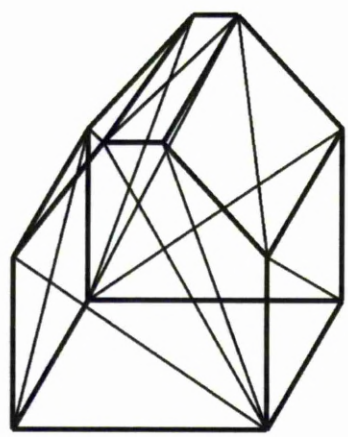


Figure 138: A sample STL file with only the edges of the triangles rendered.

Figure 139 shows regions that the STL file will be divided into. These will be a variable number of cells in the x, y, and z axis that collectively completely contain the STL file.

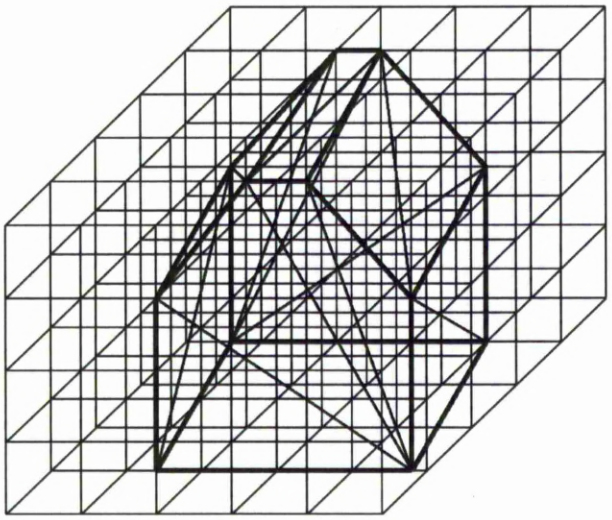


Figure 139: The sample STL file shown in figure 138 within a space broken down into many regular, cuboid boxes.

Figure 140 shows a single triangle from the file that will be processed in the following examples.

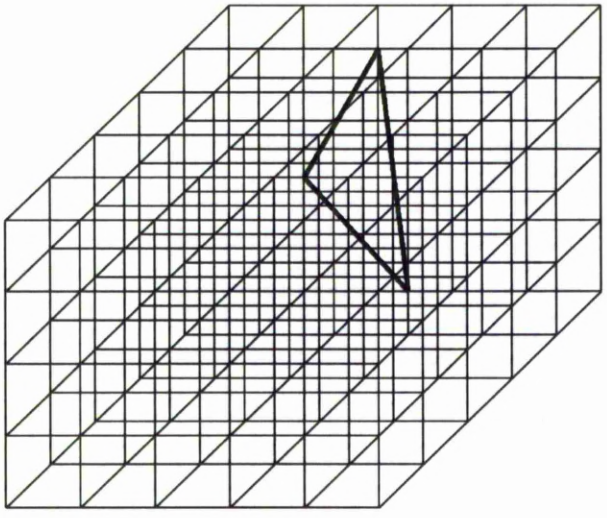


Figure 140: The sample STL file shown in figure 138 with only one triangle shown.

Each one of these triangles is taken in turn and the regions that the triangle belongs to are identified by finding which y axis columns the triangle intersects in a x-z axis projection then repeating for each axis. The x-z projection is shown in

Figure 141. This x-z projection shows the triangle intersecting five boxes. The triangle may be intersecting the vertex in the centre of the four boxes to the upper left of the image. If this is the case the triangle would be considered to belong to all four boxes.

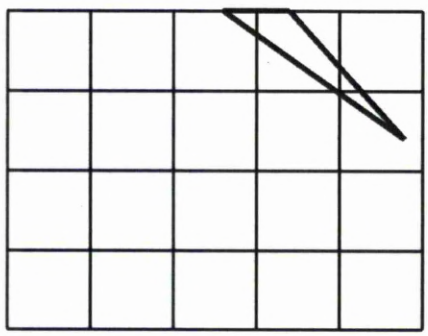


Figure 141: The triangle from figure 140 shown in the x-z plane.

In the y-z projection shown in Figure 142 the triangle appears to be intersecting an edges between boxes. If this is the case then the triangle will be considered to belong to the pillars on both sides of the edge.

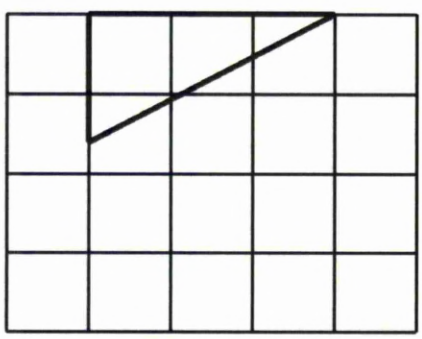


Figure 142: The triangle from figure 140 shown in the y-z plane.

Once the x-y projection is complete the 3D boxes that the triangle belongs to is found by the intersection of the three sets of pillars.

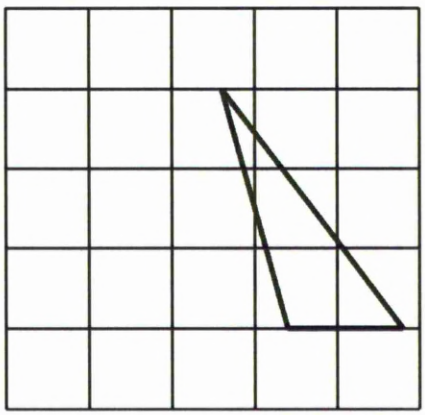


Figure 143: The triangle from figure 140 shown in the x-y plane.

Once all the triangles are processed the total number of triangles that are checked for intersection during a subsequent ray check is substantially reduced. The triangles listed in the regions that the ray passes through are checked for intersections with the ray rather than the whole list of triangles contained within the CAD file. An additional time saving is granted if a region has no triangles and all eight vertex describing the region are either inside, or outside the CAD. In this instance any elements that falls completely within said region need not be checked at all. The element is within when all vertex are within the CAD. If the eight vertex do not belong to the volume enclosed by the CAD then any element completely inside the region is scrapped as it is outside the volume.

6.1.9 Allowing Free Form Topology Generation

Following the move from direct-to-slice to full 3D geometry generation there was a requirement to create new cell topologies quickly. This was to aid the development of parallel research in to applications for lattice structures created using the SLM equipment in heat exchange and biomedical implantable components.

This necessitated the development of a clear method for describing the lattice structure's topology. This required code to read the new descriptions and build the lattice structures based on the description's instructions.

6.1.9.1 Defining the Grid:

The lattice structures built using the description and program given in the following few pages are all based on a unit cell defined by eight vertices, which describe six faces, such as a cube. The width, depth, and height need not be equal.

The first step in the lattice generation was to create a cloud of points that described the vertices of all the units on to which the additional points that are needed to create the lattice, and the truss elements will be assembled onto.

Name	Definition	Example
Number of Cells	x cells, y cells, z cells	3, 3, 3
Extents of Lattice	x min, x max, y min, y max, z min, z max	0.0, 10.0, 0.0, 10.0, 0.0, 10.0

Table 7: Method for specifying the density and size of a regular lattice structure.

The values in the list "Number of Cells" are positive integers, and the values in the list "Extents of Lattice" are decimal values.

The custom lattice topology is described in a unit cell of unit length in each axis as shown in Figure 144. This unit cell is then scaled to suit the cell size in the generated lattice. The eight vertexes are in the locations and order shown in Table 8.

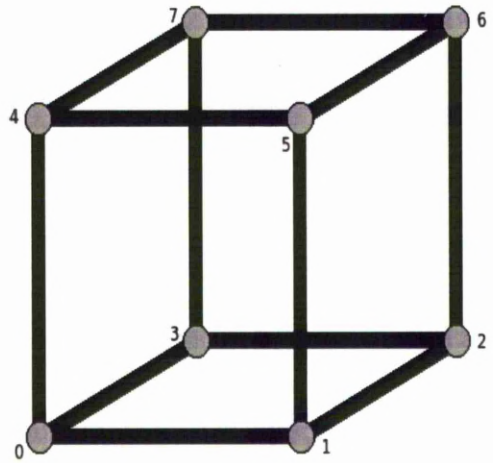


Figure 144: Visualisation of the unit cell showing the positions of the reference vertexes.

ID	Co-ordinate
0	[0,0,0]
1	[1,0,0]
2	[1,1,0]
3	[0,1,0]
4	[0,0,1]
5	[1,0,1]
6	[1,1,1]
7	[0,1,1]

Table 8: Co-ordinates that describe the unit cell.

Using the IDs defined in Table 8 the edges are described as in Table 9. Figure 145 shows the four x edges. These edges will be shared with eight neighbouring cells in the same y-z plane.

Name	Line
X Edge 0	[0,1]
X Edge 1	[3,2]
X Edge 2	[7,6]
X Edge 3	[4,5]
Y Edge 0	[0,3]
Y Edge 1	[1,2]
Y Edge 2	[5,6]
Y Edge 3	[4,7]
Z Edge 0	[0,4]
Z Edge 1	[1,5]
Z Edge 2	[2,6]
Z Edge 3	[3,7]

Table 9: Showing how the x, y, and z edges are formed from the reference vertexes.

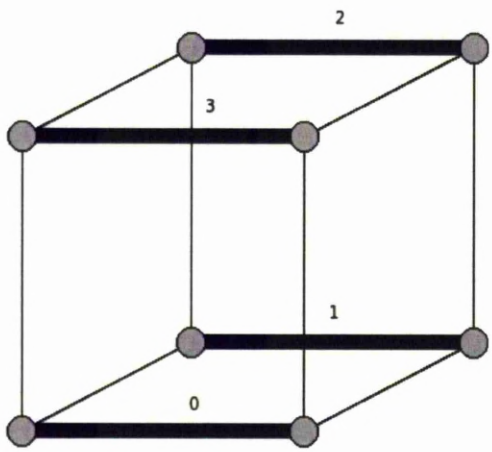


Figure 145: Diagram showing the order and position of the x edges.

Figure 146 shows the four y edges. These edges will be shared with eight neighbouring cells in the same x-z plane. Figure 147 shows the four z edges. These edges will be shared with eight neighbouring cells in the same x-y plane.

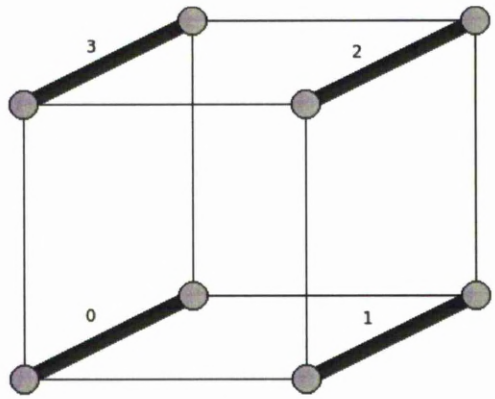


Figure 146: Diagram showing the location and order of the y edges.

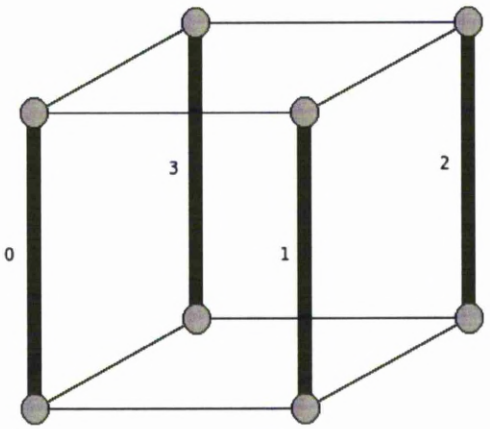


Figure 147: Diagram showing the location and order of the z edges.

The face pairs 4-2, 1-3, and 0-5 are defined as in Table 10. Figure 148 shows the faces labelled 4 and 2. These faces will be shared with two neighbouring cells which share the y and z values.

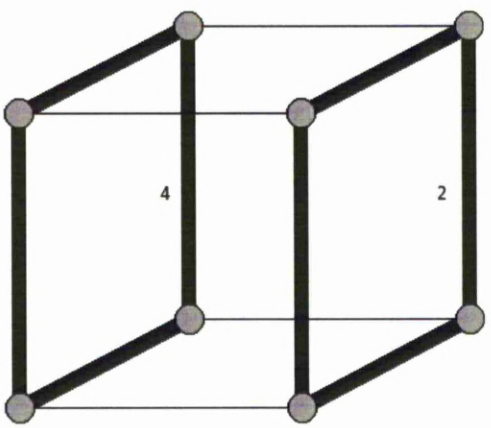


Figure 148: Diagram showing the location of the 4 and 2 faces.

Name	Face
Face 0	[0,1,2,3]
Face 1	[0,1,5,4]
Face 2	[1,2,6,5]
Face 3	[3,2,6,7]
Face 4	[0,3,7,4]
Face 5	[4,5,6,7]

Table 10: Showing how the 6 faces are formed from the reference vertexes.

Figure 149 shows the faces labelled 1 and 3. These faces will be shared with two neighbouring cells which share the x and z values. Figure 150 shows the faces labelled 0 and 5. These faces will be shared with two neighbouring cells which share the x and y values.

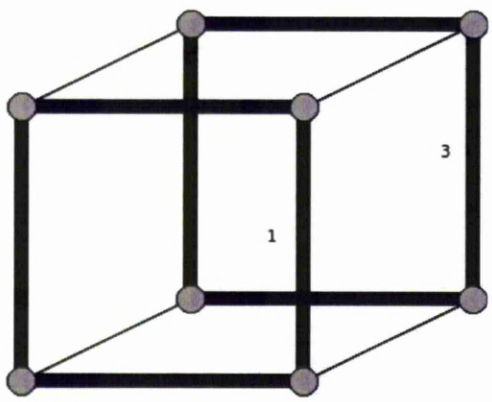


Figure 149: Diagram showing the location of the 1 and 3 faces.

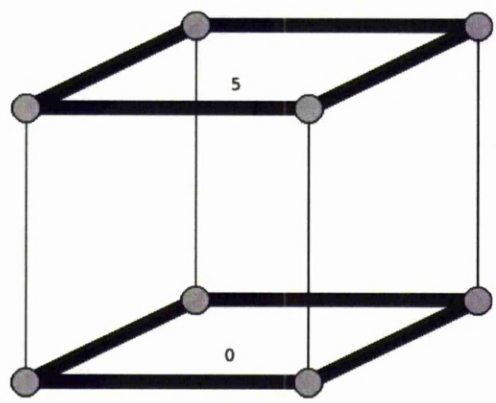


Figure 150: Diagram showing the location of the 0 and 5 faces.

The numbering custom for the cells starts at the cell with the minimum x,y,z values with an ID of 0 and progresses along the x-axis then when all cells are complete in that y-axis row it adds one in the y direction then continues as before. Once all the cells have been numbered on the current z layer it progresses to the next.

Figure 151 shows the numbering convention. The figure details the order in which

the additional points belonging to individual cells will be added to the whole lattice structure. As data is created for a cell it is recorded in data objects discussed later. If a point or link is created by a cell which is shared with a neighbouring cell the relevant information is inserted into the data object before the additional points are created in the neighbouring cell. This prevents the creation of duplicate points.

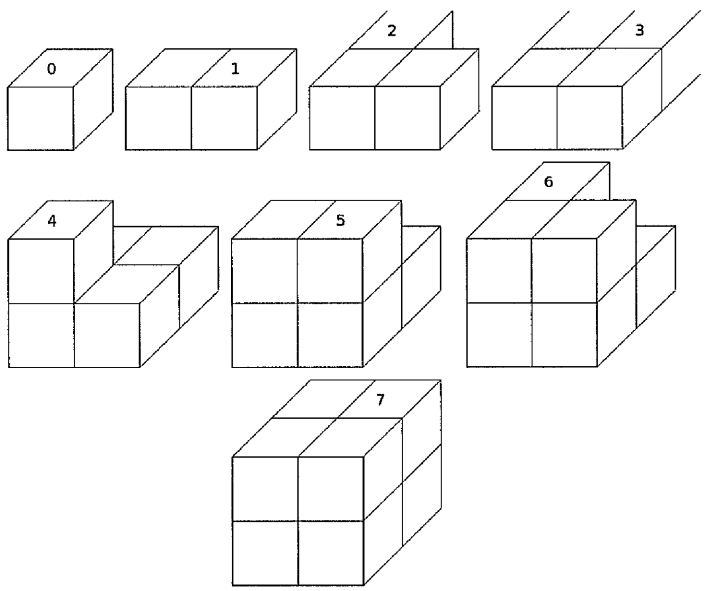


Figure 151: Assembly order for creating the grid, adding the points, and inserting the truss elements

Cells can be identified that do not require any lattice and these are subsequently missed during the generation of the lattice structure. The main use for this function is to speed up the creation of lattices that are bound by a 3D triangular surface mesh. If a cell is known to be outside the surface then it will not require lattice data to be created for it. To use this function a list of positive, integer values referring to the cells to be missed is needed.

Miss cells	[0,1,2,3,5,6,7,8,9,11,15,17,18,19,20,21,23,24,25,26]

Table 11: Example list of missed cells.

The list of missed cells in Table 11 cause the cells to be missed from the creation of Figure 152 resulting in Figure 153. The location of the missing cells is considered by the code when the additional points and elements are created for each cell.

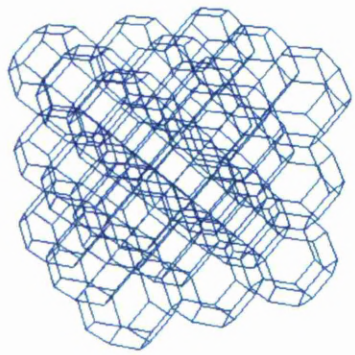


Figure 152: Example truncated octahedral lattice structure with 3 cells in each axis.

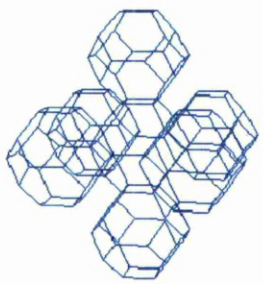


Figure 153: Lattice with missed cells.

6.1.9.2 Adding the Points:

Once the grid and miss list are fully defined the program progresses to the next stage. Points that are required in addition to the eight vertexes of the cell are added where they are needed in order to build the repeating topology for the lattice in a later stage. These additional points are split into three main categories two of which are split into sub categories. These are as follows:

1. Shared Edge Points;
    i. on x-edges,
    ii. on y-edges,
    iii. on z-edges.
2. Shared Face Points;

i. on faces 4-2,

ii. on faces 1-3,

iii. on faces 0-5,

3. Non Shared Volume Points.

It is essential to split the points into these categories so that as they are being added to the main list of points the code does not create multiple points on the same locations where neighbouring cells share points. If structures were randomised with such defects the structure would become disconnected. This would cause a proportion of the structure to fail.

For example Table 12 shows how additional points that are shared on each of the x-edges need only to be specified once and the program adds the remaining points by adding appropriate distances in each axis.

Edge	**Specified point for Edge 0**	**Code created for Edge 1**	**Code created for Edge 2**	**Code created for Edge 3**
X	[0.2, 0, 0]	[0.2, 1, 0]	[0.2, 1, 1]	[0.2, 0, 1]
Y	[0, 0.2, 0]	[1, 0.2, 0]	[1, 0.2, 1]	[0, 0.2, 1]
Z	[0, 0, 0.2]	[1, 0, 0.2]	[1, 1, 0.2]	[0, 1, 0.2]

Table 12: Demonstrating automatic generation of shared points across x, y, and z edges in a cell.

The operations as detailed on the unit cell are carried out before the co-ordinates are deformed to fit the specific cell. The user specifies the co-ordinate for the Edge-0 for each axis as specified in Table 9 and the code creates the points for edges 1, 2, and 3. As can be seen in Table 12 one cell length is added in a specific axis to move the points to the other edges. It must be noted that points shared on the edges need not be co-incident with the specified edges, they will still have one

cell length added as appropriate as in Table 12 so the code created points will not be on the edge 1, 2, or 3 either. This is so the repeating unit remains the same and links to adjoining cells without the need for an additional conjunction cell topologies.

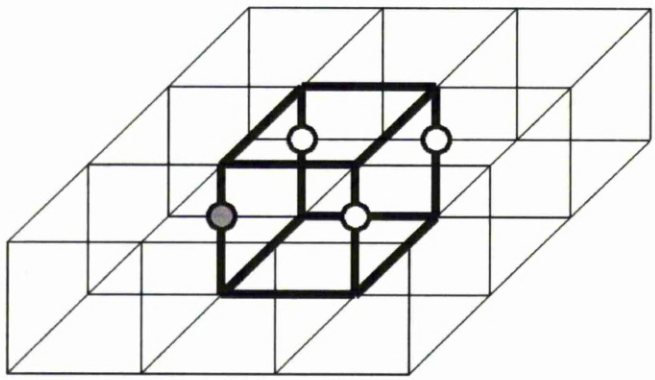


Figure 154: Diagram showing the effect of sharing a point on the y axis edge.

The consequence of sharing a point on the z axis (detailed in Figure 147) is illustrated in Figure 154. Using the cell numbering convention shown in Figure 151 the cells in the example range from 0 through to 8, with the centre cell being number 4. The grey point is a point added to the cell as a shared point on the y-axis.

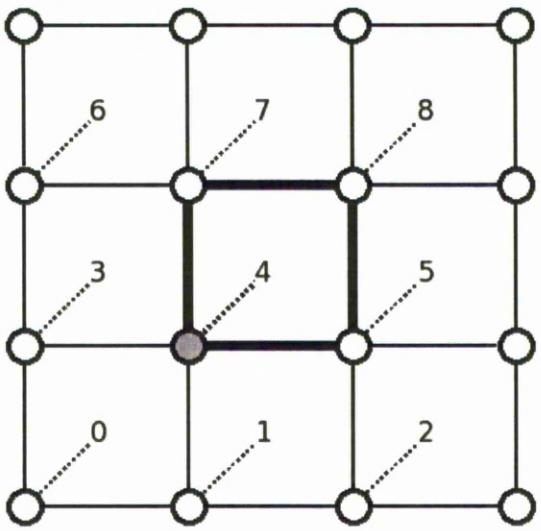


Figure 155: Diagram showing the effect of sharing a point on the y axis edge from above the cells with cells labelled.

Due to the same point being present on edge z-0 of cells 5, 7, and 8 it would appear on the z edges 1, 2, and 3 respectively on cell 4. Due to this if it is the intention for the point to be shared on the z axis, it's position need only be referenced once for the points location on edge z-0 and the additional points will generated if necessary, or if they already exist be made available for the subsequent stage of the lattice generation where the links are added to the structure. In the example given the grey point would not have had to be created as it would have already have been created by the previous cell, 3. During the point creation for cell 4 in respect to the shared points it would only need to create the points for edges 1-3. As the previous cell has no cell sharing its points to the left of it, it would have to create all the points.

The case is much the same for points shared on faces but is simpler in that there is only a need for the program to create one new point rather than three.

Face Pair	Specified point for Face 0	Code created for Face 1
4-2	[0, 0.5, 0.5]	[1, 0.5, 0.5]
1-3	[0.5, 0, 0.5]	[0.5, 1, 0.5]
0-5	[0.5, 0.5, 0]	[0.5, 0.5, 1]

Table 13: Demonstrating automatic generation of shared points across 4-2, 1-3, and 0-5 faces in a cell.

In the example given in Table 13 the user specifies a point in the centre of the first face and the code creates a point one cell length across, which in this case creates a point on the centre of the opposing face.

The effect of sharing the point listed in Table 13 for face pair 4-2 is shown below in Figure 156. The point is described once and the computer makes the point on the subsequent cell available for the trusses to link to without a need to explicitly create the additional point.

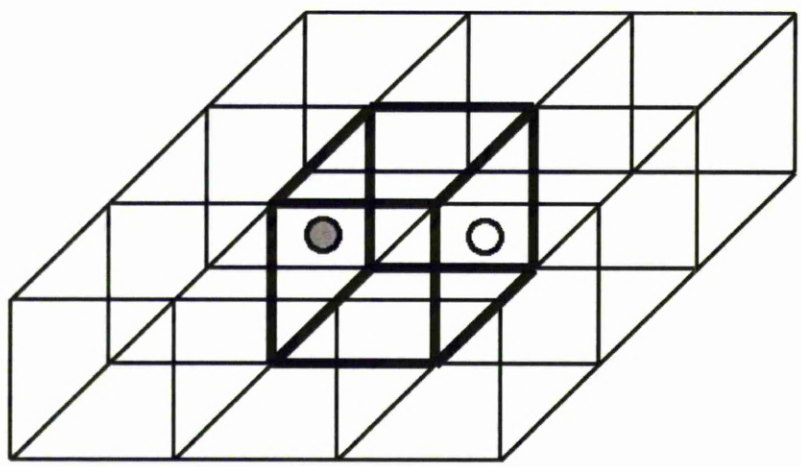


Figure 156: Diagram showing the effect of sharing a point on the 4-2 face pair.

The sharing of points on the face pairs is further detailed in Figure 157. This shows the sharing of a 4-2 face pair point. The grey point is the point created with the user specified co-ordinate for cell 4, and the white point belonging to cell 5 is made available for the creation of trusses on cell 4.

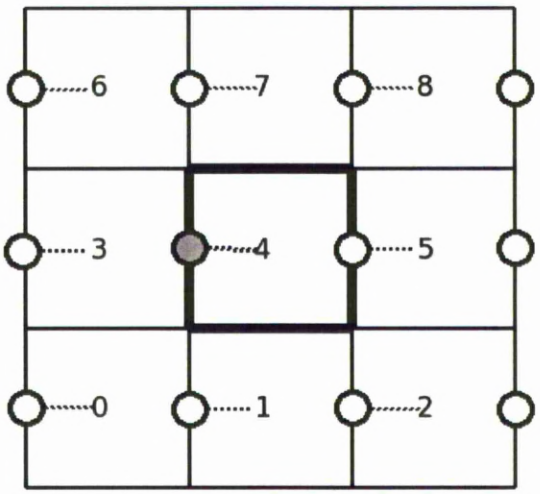


Figure 157: Diagram showing the effect of sharing a point on the 4-2 face pair from above the cells with cells labelled.

6.1.9.3 Adding the truss elements:

At this stage all the points have been created and added to a main point list. There is a data object associated with every cell that has elements to be added to it,which gives the IDs of points that the elements are to be linked to, to give the topology for the cell.

The elements are split into similar categories as the points with the addition of one category which enables elements to be created which link points from different cells which are not necessarily direct neighbours. These categories are as follows:

1. Shared Edge Elements;
    i. on x-edges,
    ii. on y-edges,
    iii. on z-edges.
2. Shared Face Elements;
    iv. on faces 4-2,

v. on faces 1-3,

vi. on faces 0-5,

3. Non Shared Volume Elements;

4. Out of Cell Elements.

The elements in the list are described using the data objects that have been created for each cell with elements in it. The complete object will be structured as follows:

Name	Base References	Additional Points
X0	[0, 1]	X0
X1	[3, 2]	X1
X2	[7, 6]	X2
X3	[4, 5]	X3
Y0	[0, 3]	Y0
Y1	[1, 2]	Y1
Y2	[5, 6]	Y2
Y3	[4, 7]	Y3
Z0	[0, 4]	Z0
Z1	[1, 5]	Z1
Z2	[2, 6]	Z2
Z3	[3, 7]	Z3
F4	[0, 3, 7, 4]	Y0 + Y3 + Z3 + Z0 + F4
F2	[1, 2, 6, 5]	Y1 + Y2 + Z1 + Z2 + F1
F1	[0, 1, 5, 4]	X0 + X3 + Z0 + Z1 + F1
F3	[3, 2, 6, 7]	X1 + X2 + Z2 + Z3 + F3
F0	[0, 1, 2, 3]	X1 + X0 + Y1 + Y0 + F0
F5	[4, 5, 6, 7]	X3 + X2 + Y3 + Y2 + F1
Volume	[0, 1, 2, 3, 4, 5, 6, 7]	X0 + X1 + X2 + X3 +Y0 + Y1 + Y2 + Y3 +Z0 + Z1 + Z2 + Z3 +F4 + F2 + F1 + F3 + F0 + F5 +Volume Points

Table 14: Example of the contents of the cell data object after all points have been added to a cell.

When the data object is created the base references are added, these are the IDs that refer to the points in the main list that make up the vertexes in the cell as

specified in Table 14. The additional points are added to the base references which refer to additional points that have been added to the edges, faces, or the volume. For instance Y3 will contain references to points added to the main point list by the code for the third y-edge. These points will been created automatically by the code from the information supplied by the user for the first y-edge.

After the data object is compiled it provides a translation between points in the individual cell and the main point list. This is best described by example as follows:

Name	Example
Additional Points	[0.5, 0.5, 0.5]
X Points	
Y Points	
Z Points	
42 Points	
13 Points	
05 Points	
Volume Points	0
X Links	
Y Links	
Z Links	
42 Links	
13 Links	
05 Links	
Volume Links	[0,8],[1,8],[2,8],[3,8],[8,4],[8,5],[8,6],[8,7]
External Links	

Table 15: Complete topology description for the unit cell shown in Figure 158.

Table 15 details a simple lattice that can be created with the addition of a single point where the line elements cross as shown in Figure 158. The single additional co-ordinate that is required is added to the additional points list and is referenced only in volume points since it is not shared with any neighbouring cells.

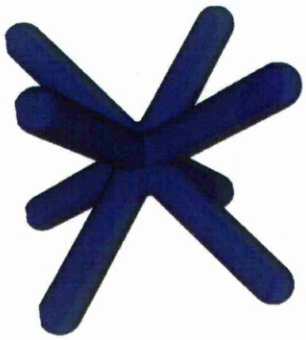


Figure 158: Octahedral repeating unit cell.

Each of the link lists with the exception of the external link list refers to the data object of the same type. For instance the values in X List would refer to points given by the IDs either X0, X1, X2, or X3 depending on which of the x-edges the code is building on.

The items in the "Volume Links" list all contain two values, which point to an ID in the Volume part of the data object for the cell in Table 15. As shown in the data object Table 15 the first eight items (0 to 7) are the IDs for the points in the main list that describe the vertexes of the cell. There are no additional points on any of the four x-edges, y-edges, or z-edges and there are also no additional points on the faces 4-2, 1-3, or 0-5. There is however one point in the volume so this ID is in the 8^{th} position in the Volume part of the cell data object. For instance [0,8] describes a line from the first vertex of the cell to the additional point in the centre of the cell.

Name	Example
Additional Points	[0, 0.5, 0.5], [0.5, 0, 0.5]
X Points	
Y Points	
Z Points	
42 Points	0
13 Points	1
05 Points	
Volume Points	
X Links	
Y Links	
Z Links	[0,1]
42 Links	[0, 4], [1, 4], [2, 4], [3, 4]
13 Links	[0, 4], [1, 4], [2, 4], [3, 4]
05 Links	
Volume Links	
External Links	

Table 16: Complete topology description for the unit cell shown in Figure 159.

Table 16 and Figure 159 show an example of the use of links that are to be repeated across the cell on the z-edges and also on two of the faces. In these cases (in a similar way to the points) the links only need to be described for the first edge or the first face and the code will multiply it out.

In this example the Z Links only has one item - [0,1] - which in this case takes the IDs of the first and the second point in the Z0, Z1, Z2, or Z3 data object lists. These are the vertices that describe the z-edges of the cell.

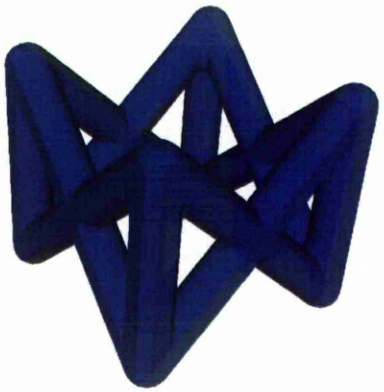


Figure 159: Repeating unit cell with diagonal and vertical elements.

The 42 and 13 link list are the same because they refer to points on that particular face. The links are built on the four points that describe the face, any additional points on the four edges, and any additional points added to the face. In the example shown in Figure 159 the first additional point is used for face 4 and the second for face 1. The additional points for the opposing faces 2 and 3 have been calculated by the code as shown in the example in Table 13.

Volume links are used where it is not required for the element to be multiplied out onto the other edges of faces.

Name	Example
Additional Points	[0.3,0.3,0.3], [0.7,0.3,0.3], [0.7,0.7,0.3], [0.3,0.7,0.3], [0.3,0.3,0.7], [0.7,0.3,0.7], [0.7,0.7,0.7], [0.3,0.7,0.7], [0.5,0.5,0.4], [0.5,0.4,0.5], [0.6,0.5,0.5], [0.5,0.6,0.5], [0.4,0.5,0.5], [0.5,0.5,0.6]
X Points	
Y Points	
Z Points	
42 Points	
13 Points	
05 Points	
Volume Points	0,1,2,3,4,5,6,7,8,9,10,11,12,13
X Links	
Y Links	
Z Links	
42 Links	
13 Links	
05 Links	
Volume Links	[8,16],[8,17],[8,20],[9,16],[9,17],[9,18],[10,16],[10,18],[10,19],[11,16], [11,19],[11,20],[12,17],[12,20],[12,21],[13,17],[13,18],[13,21],[14,18], [14,19],[14,21],[15,19],[15,20],[15,21]
External Links	[18,[1,0,0],20],[19,[0,1,0],17],[21,[0,0,1],16]

Table 17: Complete unit cell topology description for the lattice shown in Figure 160.

Table 17 makes use of the links to external cells. Note that if neighbouring cells are included in the miss list links will not be made to these cells. Links are not made to cells beyond the grid of cells described in step 1.

Each item in the external links as shown in Table 17 has three items. The first is an instruction to use the ID from the 18 position in the Volume part of the data object for the current cell. The second item instructs the code which cell to link to relative to the current cell, for the first link one cell over in the x-axis none in the y and z-axis. The third item instructs the code to use the ID from the 20 position in the Volume part of the data object for the cell which is to be linked to.

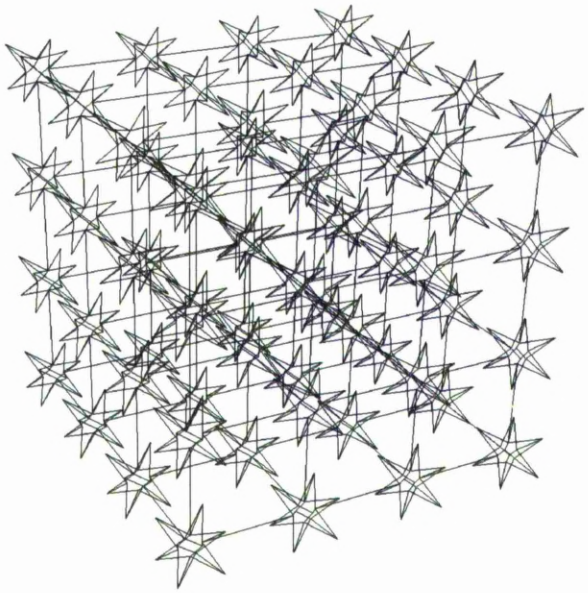


Figure 160: Example Lattice making use of external links.

Links have been made in this example to cell one across in the x-axis and one across in the y-axis and one above in the z-axis. In all cases apart from border cases the cells will also have links to one below in the z-axis, one less in the y-axis and one less in the x-axis as these would have been created when the links for these previous cells were created.

6.1.9.4 Adding Surfaces to the Topology

Surfaces can also be added to the topology of repeating cells as shown in Table 18. These are processed slightly differently to the lattice structures in the sense that the slicing of the structure results in two different geometry types being added to the slice file. These geometry types – a point sequence and a contour – are controlled by different processing parameter sets. This avoids the problem discussed earlier that led to the failure of the original lattice structures. Separating the long vectors (which are a result of slicing the surfaces), from the spots (from slicing the element lattice structures), gives the user the capability to optimising

process parameters for both without having to compromise to suit both long and short scan vectors. This was possible using the standard control software supplied with the machine.

Name	Example
Additional Points	[0.5, 0.5, 0.2],[0.8, 0.5, 0.5],[0.5, 0.5, 0.8],[0.2, 0.5, 0.5]
X Points	
Y Points	
Z Points	
42 Points	
13 Points	
05 Points	
Volume Points	0,1,2,3
X Links	
Y Links	
Z Links	
42 Links	
13 Links	
05 Links	
Volume Links	[0,8],[1,8],[2,8],[3,8],[1,9],[2,9],[5,9],[6,9],[4,10],[5,10],[6,10], [7,10],[0,11],[3,11],[4,11],[7,11]
External Links	
Shared Tris 42	
Shared Tris 13	
Shared Tris 05	
Shared Tris Vol	[8,9,10],[8,10,11]

Table 18: Complete topology description of lattice with surface elements.

Table 18 details the additional lines of information to create the repeating topology shown in Figure 161. Although not investigated during the course of this research this could be developed to create SLM models of lattice structures with thin walls rather than trusses such as some of the open cell foam structures. The surfaces could also be utilised where it is of benefit to restrict the directions in which a truss is likely to fail by buckling, or perhaps in a further investigation so

see if it effects the passage of electromagnetic radiation.

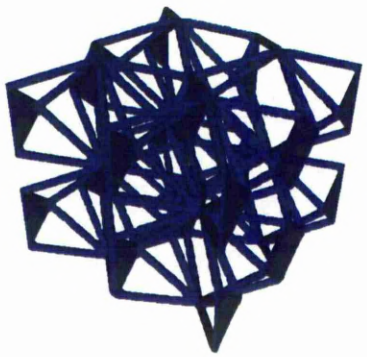


Figure 161: Lattice structure with surface elements

6.1.10 Low Angle Links

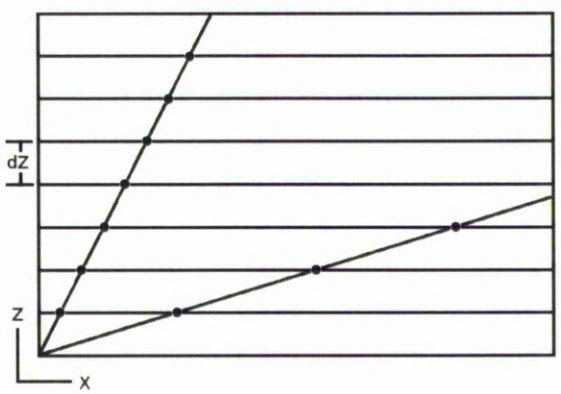


Figure 162: Detailing increase in horizontal separation of intersection points on subsequent layers as link angle from horizontal reduces.

It was observed during the initial lattice structures tests that the lower the angle the link is from the X-Y axis plane the lower the chance of the link being successfully built. Figure 162 shows two links with differing angle and their intersection points with a sequence of layers with a separation of dZ. As the link angle reduces the horizontal separation of two consecutive intersections also increases. This results in less molten material being processed per unit length of

link. Less molten material means that there is less solidified material in the final link, and this causes the link to be narrower. In addition to this when each intersection is processed if the horizontal separation is too great, the molten material will not wet onto the previously created structure. If the melt does not wet onto the previous structure then it will solidify separately to the rest of the link and be wiped away on the next pass of the recoater. This results in a failed link.

6.1.10.1 Variation of Processing Parameters

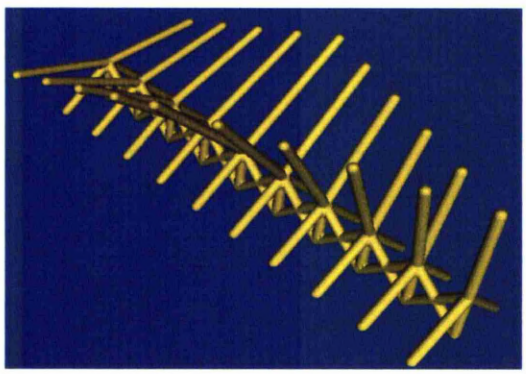


Figure 163: Computer rendering of link angle test part.

Link Number	Link Angle
1	90.00
2	78.69
3	63.43
4	59.04
5	51.34
6	45.00
7	38.66
8	30.96
9	21.80
10	11.31

Table 19: Link angles of initial set of low angle link trials.

6.1.10.2 Using Scan Vectors

The initial two tests on low angle links (detailed later) showed that the probability of low angle links building correctly could be increased with changing the processing parameters, primarily increasing the laser power.

It would be preferential to have the processing parameters of the intersection points of the slice plane and truss elements set according to the trusses angle to avoid the problem of near vertical links with increased laser power causing wear to the wiper blade. At this stage in the experimentation this was not a possibility due to the limitations of the file formats that control the SLM equipment. The problem was that there is only one process parameter set in the slice file format for the geometry type that can handle the point sequence style intersection data for the trusses.

Scanning the intersection points with vectors instead of single points was investigated to resolve this issue and to further reduce the angle at which low angle links could be built. These vectors where aligned to the direction of the

truss. Figure 164 shows the intersection points of many layers stacked to show how a vertical link with three angled links are processed with single exposures of the laser at each intersection point.

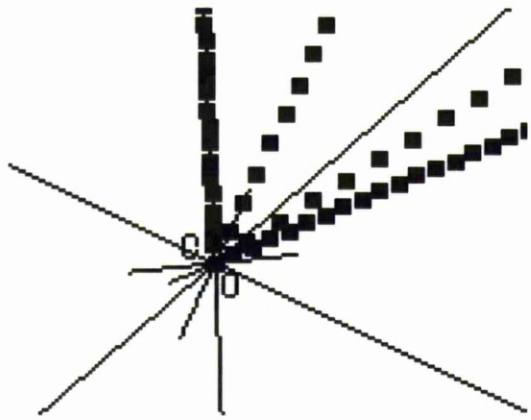


Figure 164: Image from build preparation software showing processing by spots.

Figure 165 demonstrated the use of vectors instead of single exposures, but only on the angled trusses.

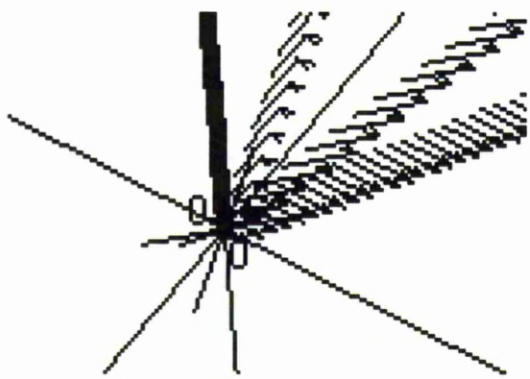


Figure 165: Image from build preparation software showing processing by spots for the vertical pillar and by vectors for the angled pillars.

In order to get a better spread of results with the vector trials the angles of the original test parts where adjusted so that the range of angles that were likely to fail covered the first through to the last link. This was so that the successfulness of the different settings was easier to distinguish between. In addition to this the lowest

angle in the structure was reduced from 11.3 to 5.7 degrees. Table 20 Shows the adjusted spread of angles for the test parts.

Link Number	Link Angle
1	45.00
2	41.99
3	38.66
4	34.99
5	30.96
6	26.57
7	21.80
8	16.70
9	11.31
10	5.71

Table 20: Adjusted link angles for follow-up trial with process parameter changes.

These link angle tests made use of the import feature to load structures into the lattice data generation software where the structures were created elsewhere and loaded and sliced to be processed on the SLM machine.

6.1.11 Horizontal Links

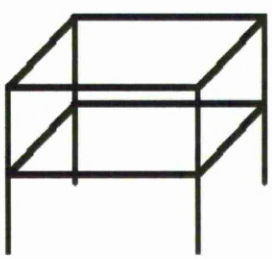


Figure 166: Horizontal link test part.

The design of the part used for the horizontal link builds is shown in Figure 166, this part has four vertical pillars on the corners of a five millimetre square, measuring 5mm in height. At 2.5mm and 5.0mm four horizontal links join the four vertical pillars. The parameter settings were controlled separately for the

pillars and horizontal links, the settings used for the vertical pillars being known to build reliably and without overbuilding.

Each of the horizontal links was split into a number of points along its length, a set distance apart defined as the point distance. Once the horizontal link was broken up, the order in which the points were scanned was defined by the scan strategy. The parts where randomly ordered on the build area to as far as possible avoid position related effects skewing the results.

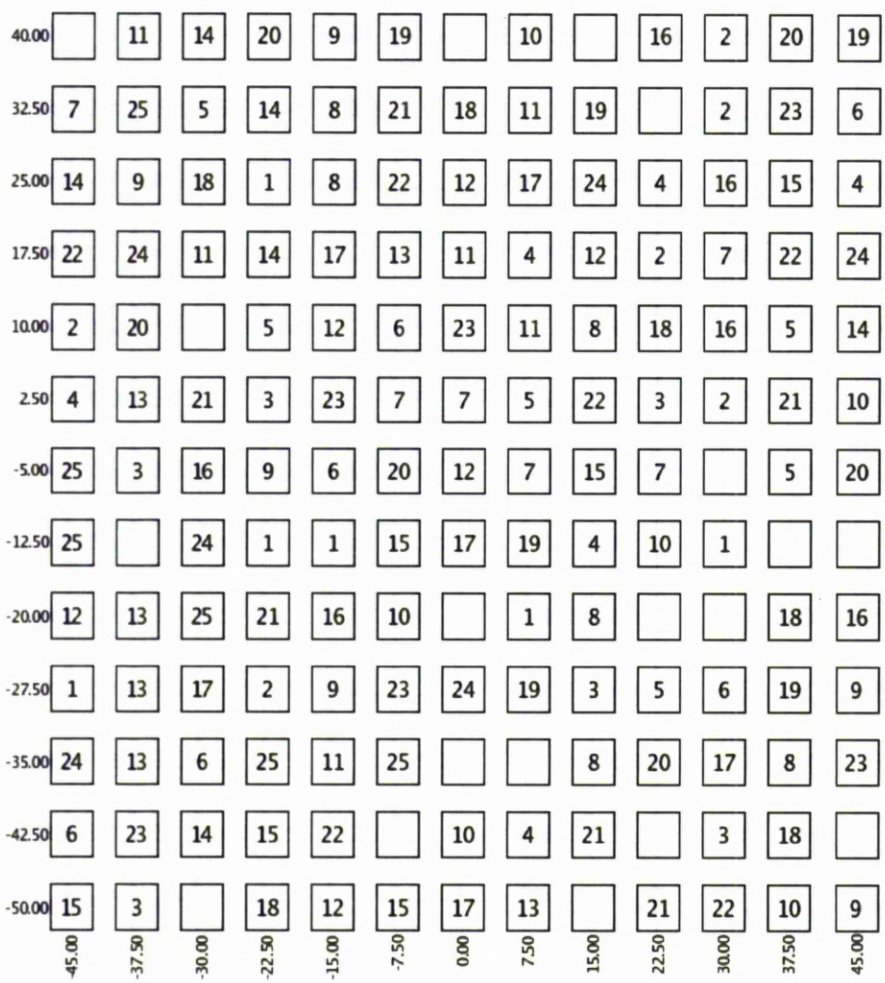


Figure 167: Part locations in a build file.

Any possible effects of position was minimised by building the same structure,

with the sample build parameters, at six randomly positioned points within the build chamber (detailed in Figure 167, axes gives bed position in millimetres.). If part position within the array had an effect on part quality, then the random positioning of the parts should reduce its significance.

6.1.11.1 Generating Single Pass Scan Strategies

As described in the introduction the Realizer system melts a line in the powder bed by targeting the laser focus spot at a series of points a defined point distance apart in sequence from the start of the line to the end. Instead of instructing the Realizer to draw a line from one point to another and allowing the software within the system to break this line up, the following five strategies were used to form the line, by prescribing to the machine the order in which the points on the line should be melted. In addition to this with the Scan Strategies 4, 5, and 6 the laser was targeted at some, or all of the points on the line twice.

Scan Strategy 1 (Figure 168) melts the first point on the line followed by the last then continues to target the laser at points on the line on progressively closer to, and on alternate sides of the centre.

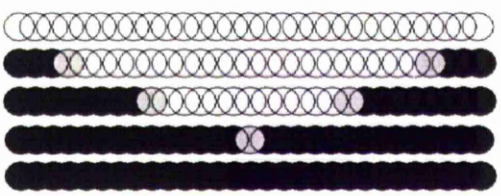


Figure 168: Scan Strategy 1

Scan Strategy 2 (Figure 167) targets the laser at the first point on the line, the point closest to a 1/3 of the distance along the line, and a final point 2/3 of the distance along the line. This process then repeats with the following points being

the second point in the line, point closest 1/3 plus 1, and the point closest 2/3 plus 1. This sequence repeats until the laser has been targeted at all the points on the line once.

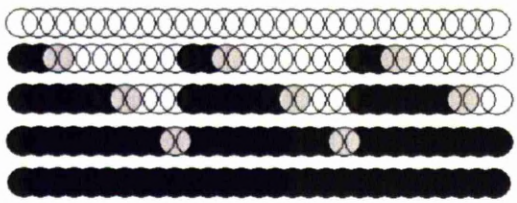


Figure 169: Scan Stratagey 2

6.1.11.2 Generating Multi Pass Scan Strategies

Scan Strategy 3 (Figure 170) is the same as Strategy 1 with an additional pass of the laser. This pass starts from the point closest to ¾ of the distance along the line and remelts the sequence of points in order to ¼ of the distance along the line.

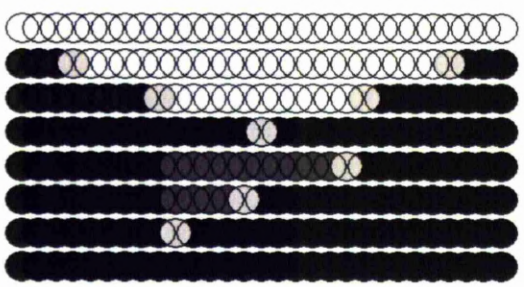


Figure 170: Scan Strategy 3

Scan Strategy four (Figure 171) is the same as Strategy 1 with an additional pass of the laser. The second pass of the laser for scan strategy four was the opposite to the first pass, the laser starting in the centre of the link and working out to the edges.

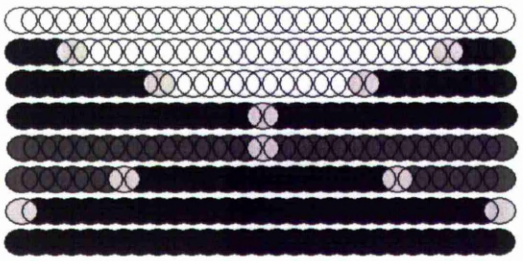


Figure 171: Scan Strategy 4

Scan Strategy 5 (Figure 172) is one pass of the laser covering the points in order from the first point on the line to the last followed by a reverse pass of the laser from the last point on the line to the first.

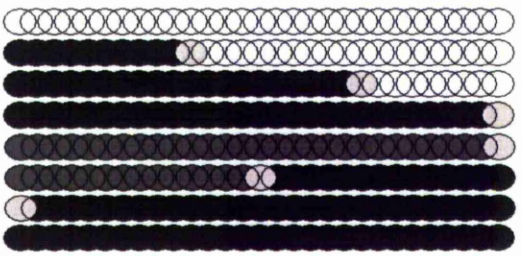


Figure 172: Scan Strategy 5

6.1.11.3 Generating Pulsed Scanning Techniques

There were significant limitations imposed on these experiments by the control software on the equipment. Primarily there was only one parameter set available to use to process the horizontal links. It is highly unlikely that the best laser and scan settings for the first and second pass of the laser are the same because the conditions are so different. On the first pass of the laser over the horizontal links the laser is aimed at powder, where as on the following pass the laser is being focused on previously processed powder - solid metal. In order to make further progress with this experiment more control of the machine is required. To achieve this the workings of the machine control system where investigated.

The software on the user interface machine was replaced with a multiple threaded and process software application suite which allows the experimental freedom of as many parameter sets as are required to perform experiments, and additional control which was not possible through the standard control software. The operation of the software is detailed in Figure 173.

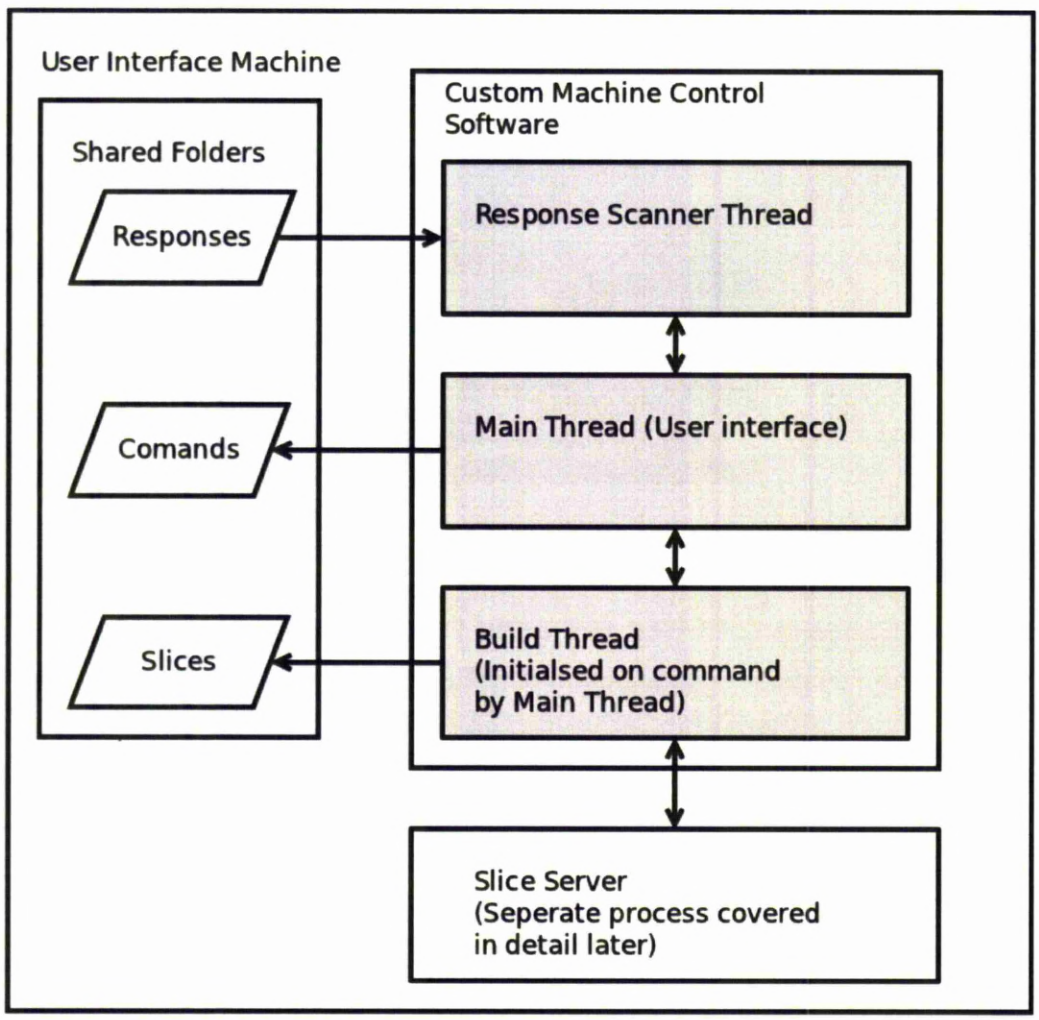


Figure 173: Flow chart detailing the replacement machine control software.

The software designed for this purpose splits the task of controlling the slave machine into two. Firstly there is passing commands to the machine. In order to be compatible with the existing hardware this is in the same manner as the

proprietary software sold with the machine. With the second part, in contrast to the proprietary software, the slice data is requested from and made by another process.

This software architecture enables the better use of multiple core processors, as the slicing software can run on a separate core (the distribution of processes over multiple cores is controlled by the operating system) and can prepare slices ahead of time while the other process is controlling the machine. This makes best use of dead time in the process while the machine is working - carrying out actions such as scanning a layer - and the main computer is mainly idle.

The machine controller software consists of two threads that are continually running while the application is alive, the main thread, and the response scanner. The main thread contains the GUI code (to allow human interaction), as well as an event queue, and a machine status object. The event queue is used for thread safe communications to the GUI code, and for instructions. Other threads can access the machine status object directly to edit or view the current status of the machine.

The main thread contains data about the state of the machine, what commands are pending response from the slave, and a queue which receives events from the response directory scanner and build thread.

The main thread also receives the user input, which varies from commands to open/shut valves and activate pumps through to starting a build process. The simple commands are passed for the slave to action via the commands directory. The user start build request causes the main thread to start the build process thread that controls the slave computer during the build sequence.

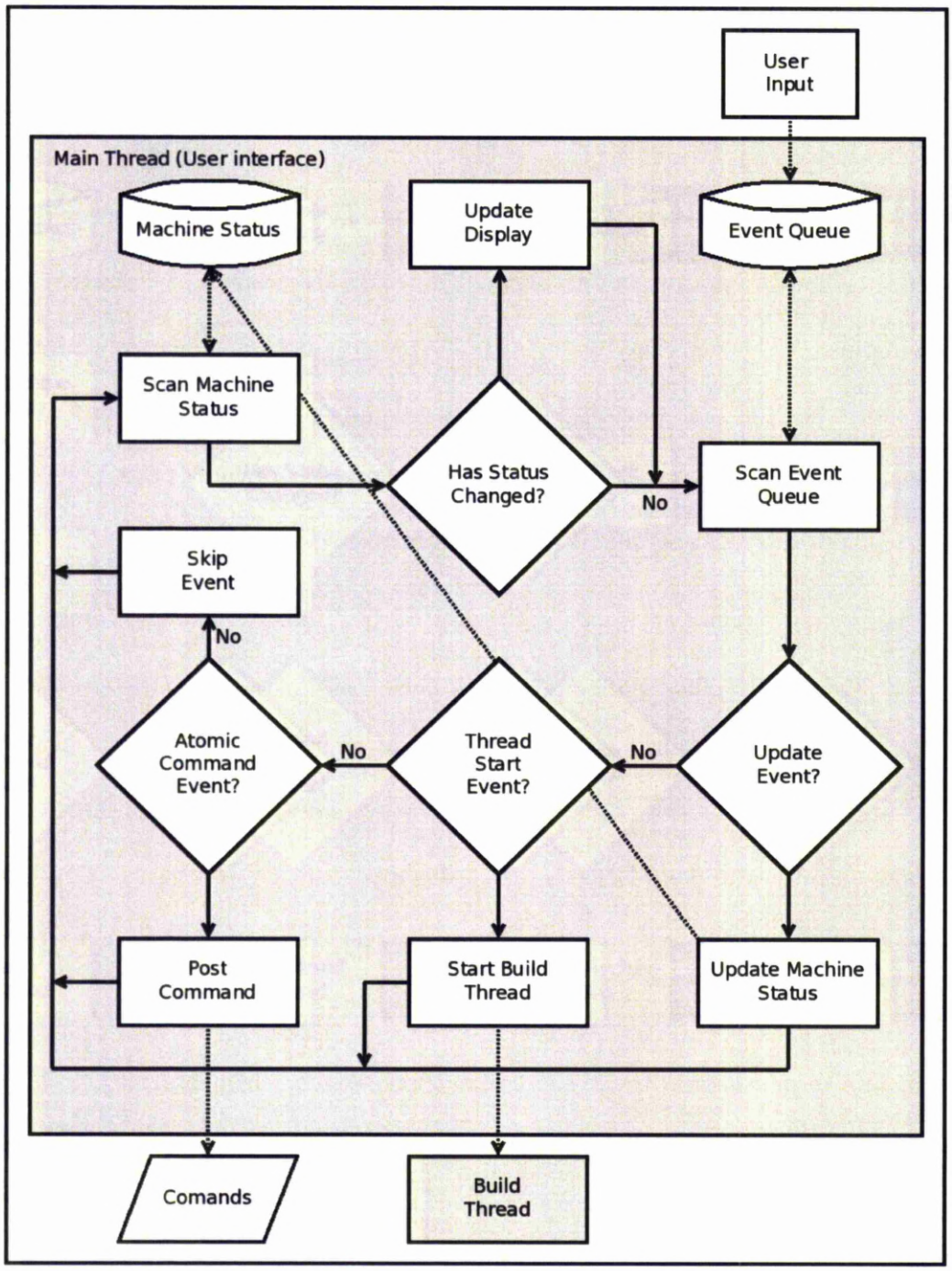


Figure 174: Flow chart detailing the main thread of the replacement machine control software.

The response scanner is a thread that removes the need for repetitive scanning of the response directory within the main thread. On receipt of a response from the Slave computer it is parsed to identify the type of response it is and split off any

data that may be attached. For example the slice request has associated data – the slice number, whereas the slice imported has no data. For those responses with data the response thread acquires a lock on the Machine Status in the main thread and updates the status with the data. After this the lock is released and an event is created and passed to the main thread that alerts the thread to the receipt of the response and it is subsequently acted on by the main, or build thread.

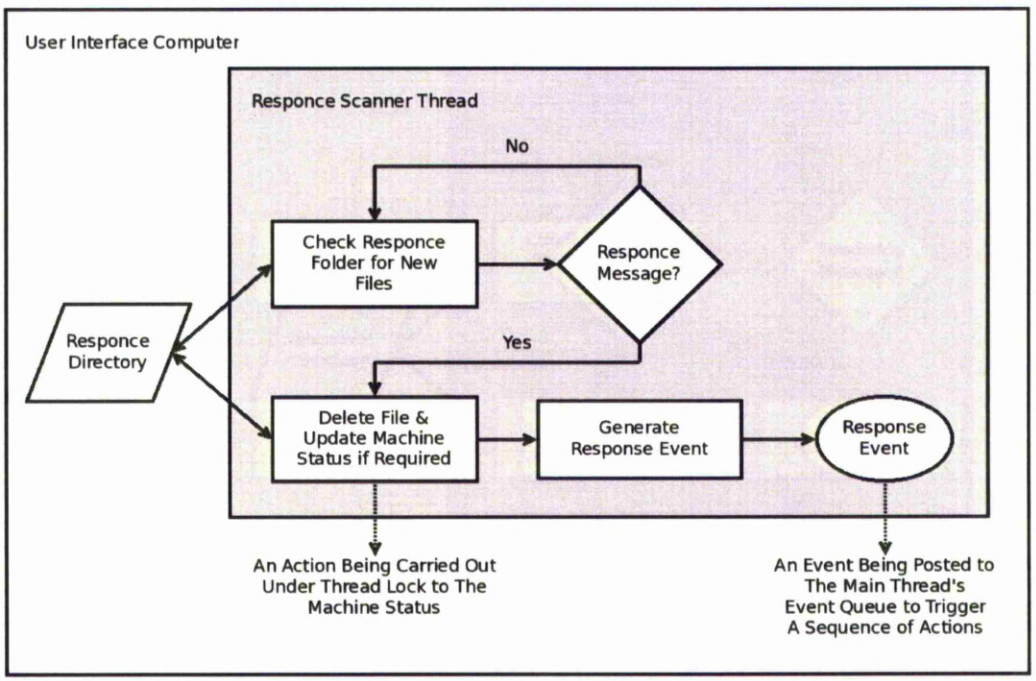


Figure 175: Flow chart detailing the response directory scanning thread of the replacement machine control software.

The build thread repetitively acquires a lock on the machine status in the main thread and checks the current status of the slave computer and what it has responded to. This is to see if it has requested a slice, waiting to fire, or the user has requested an action – for example a pause process. It is this thread that ensures the slave gets appropriate commands when the slave is ready for them.

Additional short lived threads are created when the control software has to wait

for something to complete. For example in normal operation when the software starts a build it will send one command, and wait for a response to confirm the machine has initialised for a build. Rather than sending the command and entering a blocking loop until the command confirmation is received a thread is started that fires an event into the main threads event handler which confirms the command has been processed and continues with the next step. These short lived threads reduce the complexity of the code, and enable otherwise blocking events to be handled without locking the user interface.

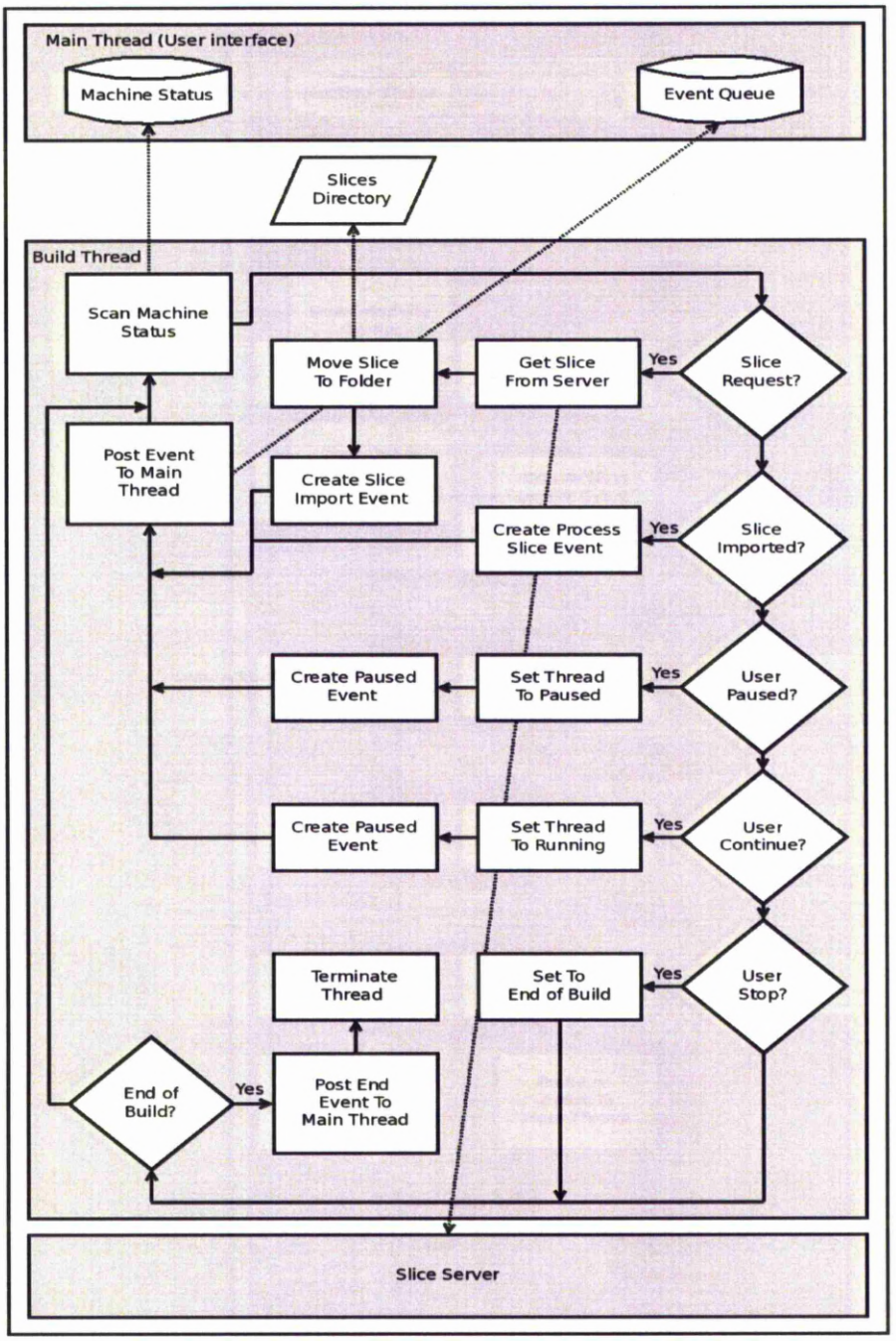


Figure 176: Flow chart detailing a build thread of the replacement machine control software.

When the slicer is activated it becomes a server sitting on a network port of a machine, in much the same manner as a web server would serve web pages over port 80, this software serves slice files when requested by the machine. When a

slice is required the machine controller makes a temporary connection to server over the network, which reads the request and responds by mirroring the request followed by the job file and the connection is closed.

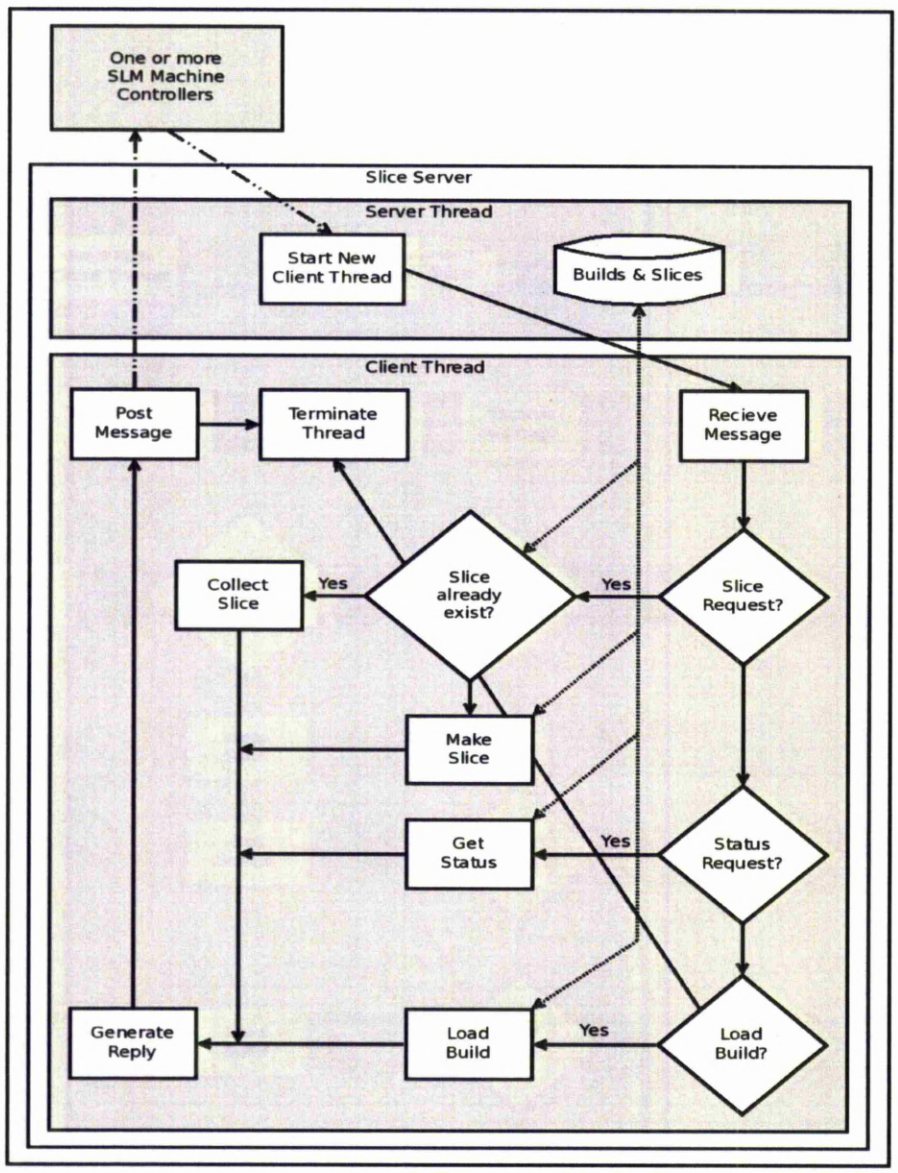


Figure 177: Flow chart detailing the slice server of the replacement machine control software.

The use of the new machine controller enabled the addition of pauses between processing spots along the length of the horizontal link. The pauses in between spots where introduced to give the just exposed molten material time to cool and

solidify. This reduces the length of the molten track, and so avoids the Rayleigh instabilities.

6.1.12 Software Development Summary

This section of the results and discussion has been focused on the development of the techniques required to create files to run the SLM equipment in order to create the lattice structures, and feed new techniques and tools into collaborative research. This work enabled the following research to be carried out which was used to validate the developed software techniques to create blocks of lattice, shaped lattice structures, the effects of randomisation, and what effect the proposed techniques to expand the processing capabilities have in terms of the limits in link angles and horizontal links.

6.2 Lattice Structures Build Tests

Unless stated all test builds discussed in this thesis were run in stainless steel 316L grade powder which had a maximum particle size of 50 microns and an average of 38 microns supplied by Sandvik Osprey Powders. The heated platform was not used at any point. The build chamber was prepared so that the system reported oxygen content of the chamber was no greater than 0.5% and the gas flow rate into the chamber was set to maintain a positive pressure of 15mBar.

6.2.1 Direct to Slice File

Lattice structures created using the open-surface supporting geometries required a very specific parameter set to enable the structures to build. This was due to the previously discussed differences in vector length within the slice data. The majority of the slice data for the structure comprised of very short scan vectors or points. Occasionally a scan vector many multiples longer would be present close to where two or more lattice elements join. The length of the shorter vectors were less than the used point distance. This resulted in the laser forming these vectors with a single laser exposure point. The longer vectors comprised of many exposure points. The point distances being used for these structures were less than one laser focus spot diameter, as is normally the case. When the scanners process a line which is broken into more than one point, each point (apart from the first) re-exposes a portion of the previous point on the line, and is subsequently partially re-exposed when the following point in the line is processed. The partial re-exposures of each point in a line which is longer than a single point distance results in a higher energy being delivered per per unit volume than when compared to a vector that is completed with just one exposure if the same

processing parameters are used. Due to this the process parameter window for the short single vectors is different to that of the longer vectors.

Parameter sets that were able to create lattice element sections that used points failed to build the elements when a long vector was processed. As both were processed with the same parameter sets the window of possible processing parameters that would successfully build the parts was reduced. Difficulty in processing these files has also been reported by Rehme [42].

The most common result of working with these files on the SLM machine is shown in Figure 178. The build tended to start successfully but fail at a point close to where many elements intersected.

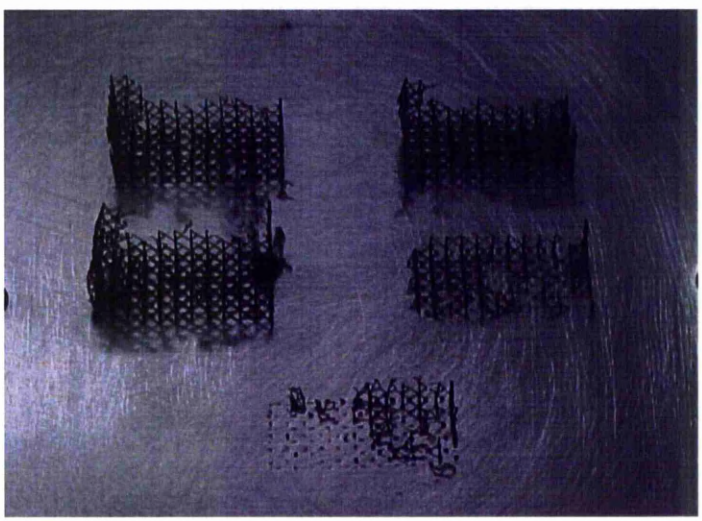


Figure 178: Failed lattice build with original files. Various attempts were made with laser powers ranging from 50 to 100W and exposure times of 200-2000µs.

Figure 179 shows the result of the first test with the new slice data from the direct to slice software. The trusses appeared much more uniform than the trusses that did build from the original open surface slice data. This was most likely due to the much improved regularity of the slice data.

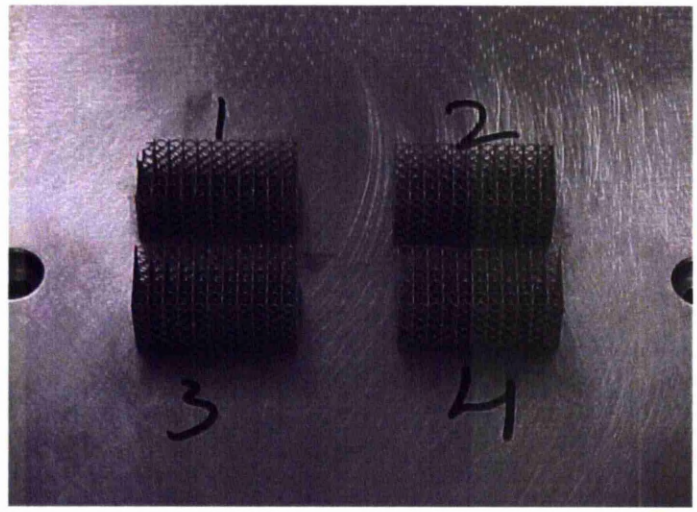


Figure 179: Successful build with new files. Built in 316L steel with a laser power of: 1) 75W 2) 100W 3) 100W 4) 100W and a point exposure time of: 1) 1000µs 2) 1500µs 3) 2000µs 4) 2200µs.

The processing window for these structures was found to be large. The results plotted in Figure 180 show tests where the laser power was raised from 75 watts through to 100 watts, and the exposure time at each point on the lattice was more than doubled from 1000 to 2200 microseconds. The graph shown is plotting the effect laser power and exposure time had on the truss radius. The result of this test was that the mean truss radius is dependent on the exposure time of the laser at each intersection, the results for increasing laser power where not as distinct.

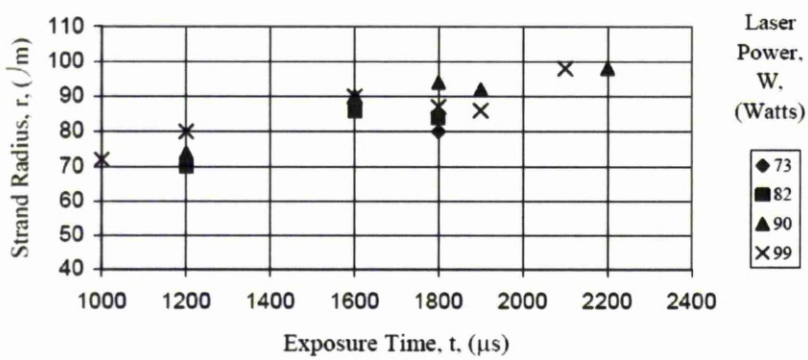


Figure 180: Strand radius verses intersection exposure time for fixed laser power. [43]

The output from the software was a single part file rather than a complete build file. This gave the possibility to mix various part files within one build. Figure 181 shows a solid skin being mixed with two different types of lattice structure. This single component comprised of four part files, one each for the lattice structure, one for the solid skin, and a final for a supporting structure. These files would have to be manually positioned within the standard build preparation software or positioned relative to each other by editing the build file in a text editor as discussed in the methodology section.

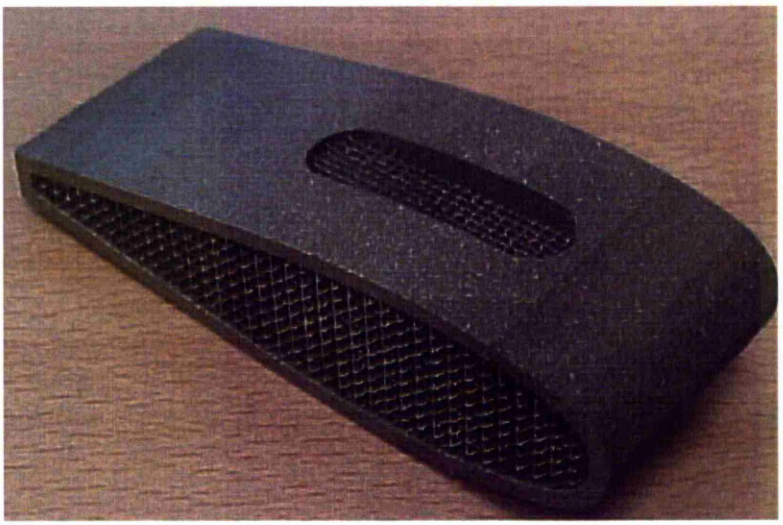


Figure 181: Example of a component with two different lattice geometries and a solid section. Lattice structure was built with 100W laser power and 1500µs exposure. The solid had a point distance of 65µm, laser power of 70W and an exposure of 250µs.

Due to the ability to mix various different parts into one build file it is possible to include solid features into the lattice structure, using the lattice as the support for the solid. Figure 182 illustrates a concept where a bearing mount is incorporated into a lattice structure.

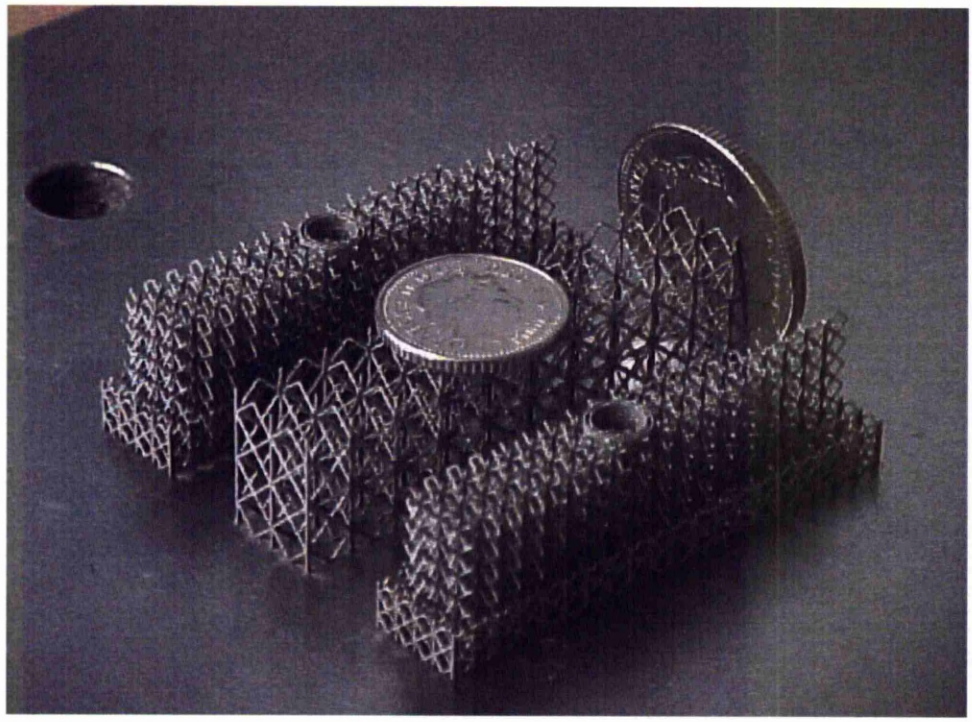


Figure 182: Detailing a shaped lattice structure with solid sections supported by the lattice structure. Built in 316L steel with a laser power of 90W and a point exposure time of 1000µs. The thin wall structure used a laser power of 90W, an exposure time of 300µs and a point distance of 65µm.

Due to the intersection data being calculated on a slice by slice basis it is possible to add continuously changing geometry into the lattice. The example in Figure 183 shows a spiral element being used instead of straight element. While this would still be possible using the 3D lattice geometry generation techniques discussed earlier, it would be more complex as the curved element would have to be broken down into sections. This technique would not work well with the randomisation of the regular structures as a technique would be required to only randomise the grid that the lattice is built upon, rather than the whole structure. Due to these difficulties and general research into lattice structures suggesting that the elements should be loaded in compression or tension to be most efficient research into this form of elements was not progressed further.

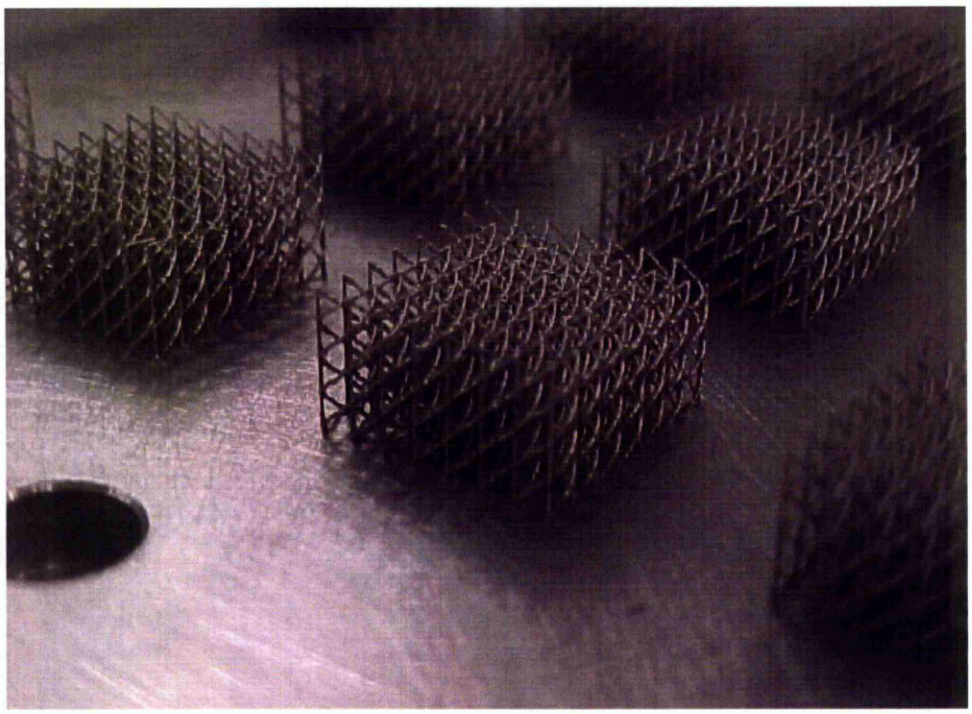


Figure 183: Alternative spiral lattice geometry measuring 17.5mm square (excluding the protrusion of the spiral elements) by 10mm tall created with the direct to slice method. Built in 316L steel with a laser power of 100W and a point exposure time of 2000µs.

6.2.1.1 Mechanical Crush Tests on Direct to Slice Lattice Structures

A number of uni-axial compression tests were carried out to assess the effect that changing cell geometry had on the crush profiles of the structures. Compression tests were carried out on a tensile testing machine (Instron 4505) equipped with a 10kN load cell. The cross head speed was set at a constant of 3mm/min for all of the tests.

Figure 184 and Figure 185 show the two standard geometries used in the initial mechanical investigation. Figure 184 shows Cell Type A, a pillar and diagonal repeating cell. This lattice topology was not specifically chosen for the mechanical properties it imparted onto the bulk lattice material. The software that created this topology was developed to satisfy the need to build the structure in

order to complete the commissioning of the experimental equipment. From this it became a default for evaluating the effects that process parameter changes had on the lattice structure because it was the only topology initially available. The combination of the vertical elements and no elements lower than 45 degrees from the horizontal x-y plane meant this was an reliable structure to create.

Figure 184: Cell type A for compression tests

Figure 185 shows Cell Type B a pillar and octahedral repeating cell. Cell type B was created to evaluate the effects of linking all the diagonal elements to the centre of the unit cell, rather than having four separate junction points.

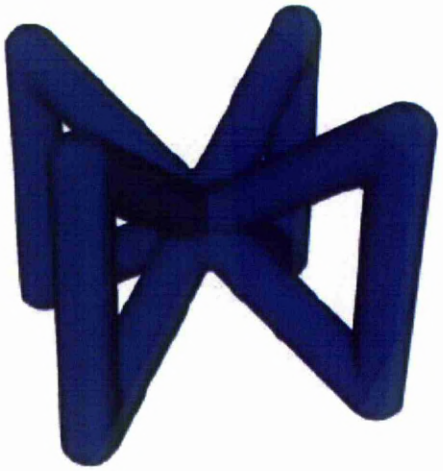


Figure 185: Cell type B for compression tests

Figure 186 shows an early build laid out so that the structures could be tested without being removed from the substrate. This avoided added additional error which could have been introduced by damage to the structures when they were removed from the substrate.

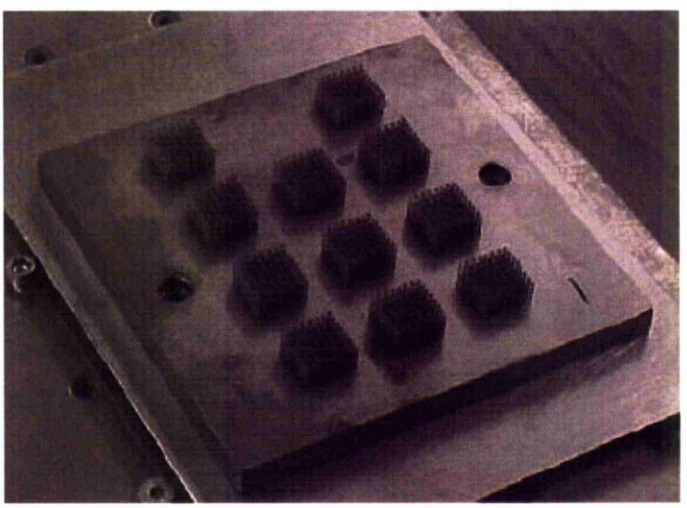


Figure 186: Compression test layout of control samples built at 1000μs exposure and 100W laser power.

Figure 187 shows the difference in strength of the two lattice cell types. These plots are 100 data points per line with each line being an average from nine

samples taken from one build as shown in Figure 186. This information was collected from the raw text file output from the mechanical test machine.

Each individual sample used the same processing parameters throughout the slice file. Comparisons between the lower angle and vertical angle elements revealed that with identical processing parameters the element diameter reduces with the element angle. This was particularly apparent for Cell Type B which has lower angle elements than Cell Type A.

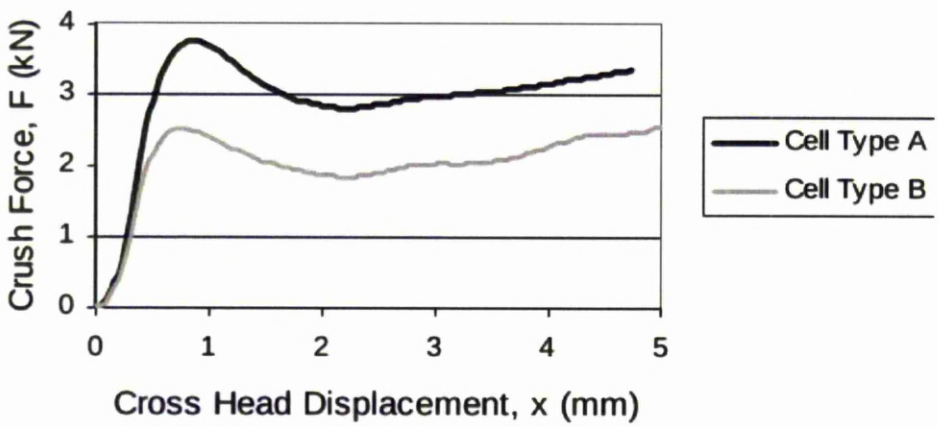


Figure 187: Crush profile of the two cell geometries with 4000µs exposure and 100W laser power. [44]

The consistency of the crush profiles of 10 identical structures (and one control sample) built in one build is shown in Figure 188. The lower line is not built at the same exposure setting as the rest. This line is the crush profile of the control sample which remained the same throughout the five builds of the cell type. The control sample was intended to check for build to build variations in mechanical properties. These control sample were built in the same location and with the same process parameters.

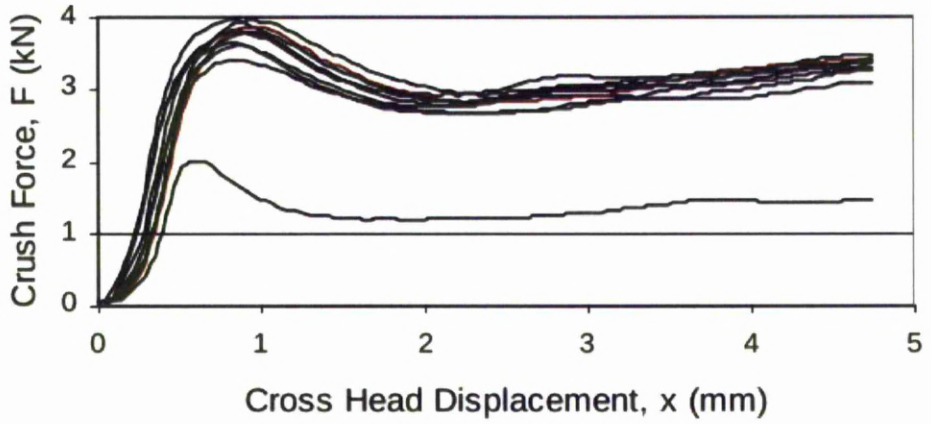


Figure 188: All crush profiles of one build of Cell Type A with 4000μs exposure and 100W laser power. [44]

When Figure 188 is compared to Figure 189 it it is clear that there is greater variation in the crush profiles of parts built in separate locations in one build than being built in the same location on different builds.

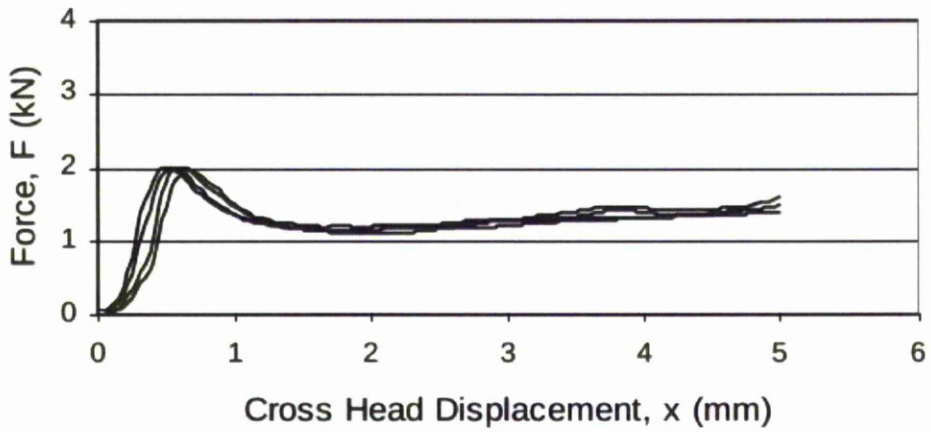


Figure 189: All control samples from five builds of Cell Type A built at 1000μs exposure and 100W laser power. [44]

6.2.2 Processing Using 3D Geometry

6.2.2.1 Regular Structures

Using the developed techniques it was possible to create highly complex lattice structures that would have been impossible to build using any other current techniques. An example of a high complexity structure is shown in Figure 190.

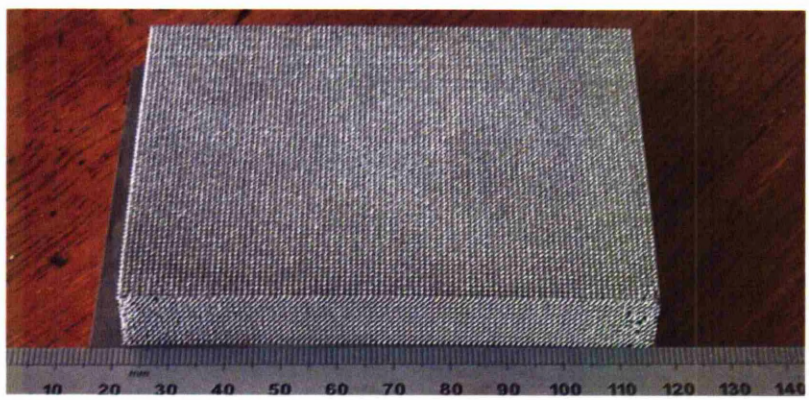


Figure 190: Regular lattice structure with 81k cells, 172k points, and 648k elements. Built in titanium with a laser power of 70W and a point exposure time of 300µs.

Wang et al 2005 [45] discussed their technique for modelling lattice structures as STL structures and the time taken for their software to create the files is shown in Table 21. Assuming the memory usage is linear with the number of elements in the structure then the above would take approximately 15 gigabytes of RAM, a quantity that is rarely seen on standard computer equipment. Beyond this the estimated time for the software to run based on the results in the table would be approximately 52hrs. However their written discussions on the creation of 20,000 element structures would suggest a build time for the creation of the STL structure would be 24hrs. This would still need to be sliced, and as components larger than the small lattice structure detailed in the methodology could not be processed in the build preparation software it is assumed something at this level of complexity

would be unworkable.

	Element#	225	480	1074	1999	3547	4662
Time (second)	ACIS	37	107	370	2400	N/A	N/A
	STL	80	222	539	991	1256	1361
Memory (MB)	ACIS	52.6	95.68	321.5	450	N/A	N/A
	STL	32	37.5	49.316	68.04	84.9	99

Table 21: Memory usage and processing time for creating STL file lattice structures.[45]

Larger structures were created and required the software source code to be reviewed for efficiencies in order to build without exceeding 1GB of RAM usage that was causing the Windows operating system on the utilised computer memory allocation problems. The largest quantity of elements constructed was in the creation of a randomised 45mm cube of lattice structure that was built with a 0.5mm cell size. This had in excess of 5.8 million elements. It is worth highlighting at this point that the utilised programming language was chosen for its ease of maintenance and development, powerful text parsing, and wide community support rather than raw processing power. It could be assumed that once limits of performance had been reached progress could be made by re-writing processor intensive sections of the code into a lower level programming language.

6.2.2.2 Random Structures

Random structures were not attempted during this research due to the large quantity of low angle and near horizontal elements. Since this decision not to attempt these structures at this stage of the research a number of collaborative projects have identified a limit to the building of lattice elements to be about 25 [18] to 30 degrees [46] to the horizontal. Using the 30 degree limit out of a

possible 180 degree range of element angles this takes a third (from -30 to 30 degrees from the horizontal) of the total range. If an even distribution of element angles is assumed this would imply a third of the structure would fail to build. Figure 191 Details this range of non-creatable links.

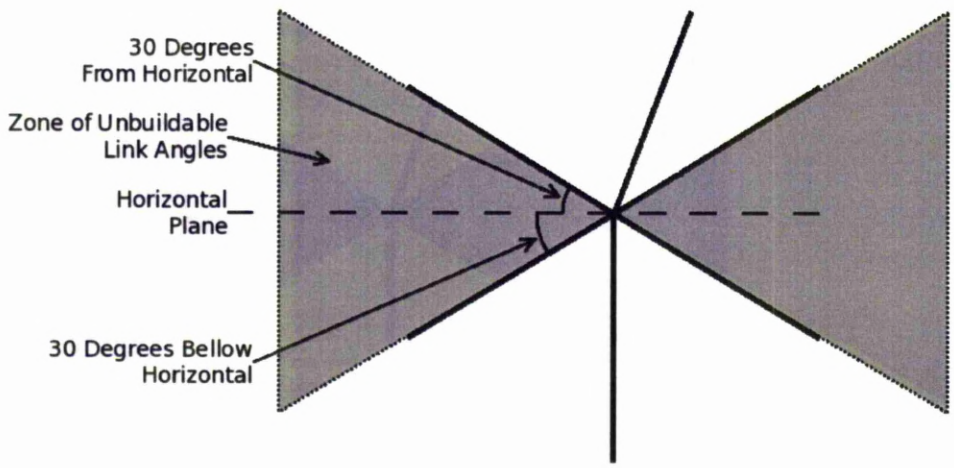


Figure 191: Detailing the range of non-creatable links from a central point. Downward pointing links are assumed to be supported at their lowest point by other lattice structure.

6.2.2.3 Polar Mapped Structures

Parts which were built to test mapping the lattice onto a polar, rather than rectangular grid are shown in Figure 190. These structures were investigated to remove regular patterns (as highlighted in Figure 193) from the surface of curved surfaces on shaped regular lattices. While it did achieve the basic aim of building a polar mapped structure it created a problem in the densification of the lattice structure towards the pole of the polar grid system. While this was apparent in the visualisation of the 3D geometry the effect is far worse in the build with the lattice structure varying from well defined around it's equator and gradually becoming dense metal at the pole.

Figure 192: Polar grid lattice structures. Built in 316L steel with a laser power of 150W and a point exposure time of 800µs.

6.2.2.4 Randomisation

The investigation and introduction of the randomisation feature was in order to control how regular the structure was, and to reduce the visual faceted appearance which is a side effect of cutting the lattice structures to a curved surface. This faceted effect can be seen in Figure 193. The upper image shows the original photographed image and lower has the the edges of the visible faceting enhanced for clarity.

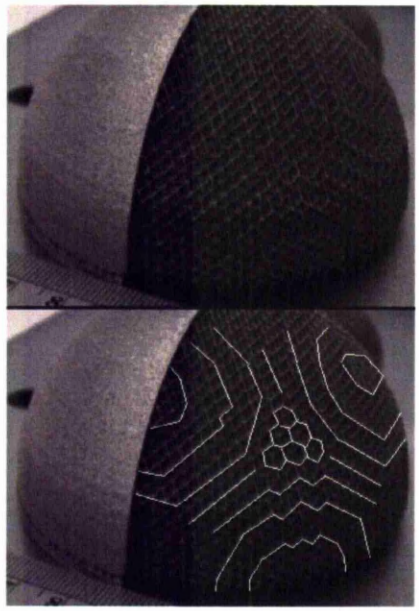


Figure 193: Detailing the faceted appearance that occurs when shaping a regular structure to a curved surface. Solid built at 65µm point distance, 250µs exposure and 90W laser power. Lattice built at 500µs exposure and 140W laser power.[18]

Figure 194 shows the visual effect of progressively greater randomisation on a regular lattice structure.

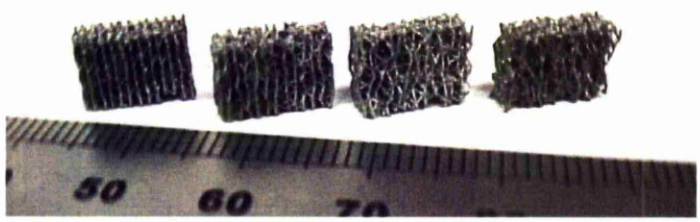


Figure 194: Effect of progressively increasing lattice randomisation. Built in 316L steel with a laser power of 150W and a point exposure time of 1500µs.

The problem of the regular pattern is seen on the two parts closest to the camera on the right of Figure 195. This shows an initial trial of the randomised structures on parts consisting of solid and lattice structures. In this case the lattice structure is intended to provide a bone ingrowth surface. Collaborative research investigated the effect of randomisation in terms of the bulk materials mechanical

properties. [46]

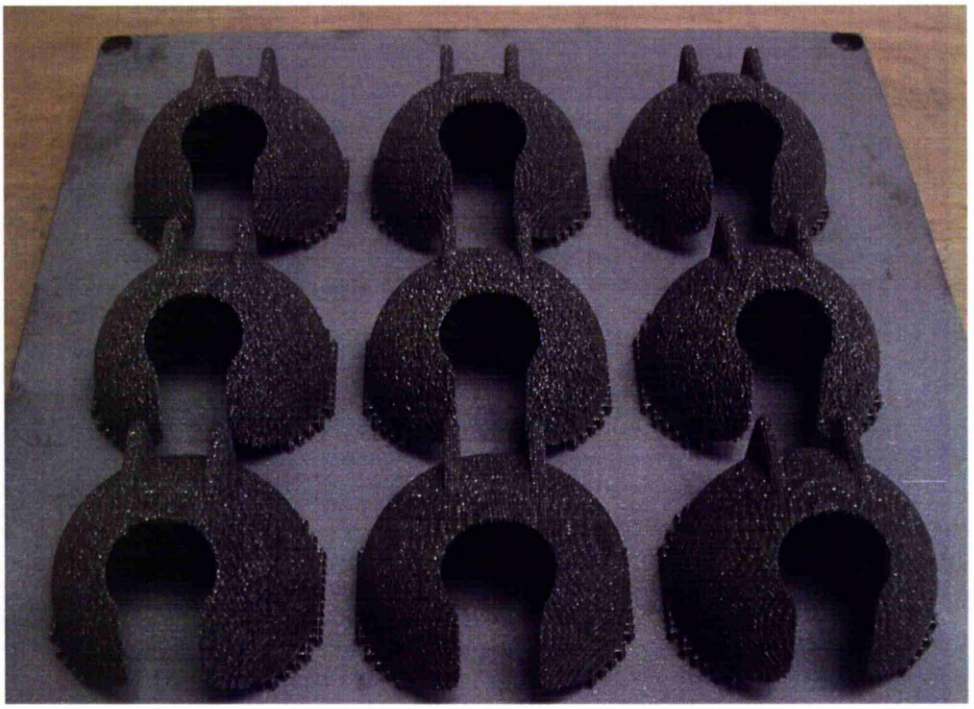


Figure 195: A range of lattice and solid components built with varying degrees of lattice randomisation. Laser power of 90W and exposure time of 300µs. Point distance was 65µm.

6.2.2.5 Supported Structures

Supporting the lattice structures was achieved through one of two methods previously described; repeating the first slice of a build a number of times, or supporting unsupported geometries after analysis of the 3D lattice geometry. Figure 196 shows an array of parts built on a support structure that directly targeted unsupported points in the lattice. This wasn't useful for solid/lattice parts similar to those shown in Figure 195 as the stresses during build caused the supports to be pulled off the substrate and fail. During lattice only builds this technique provided a useful way of creating a minimal support structure to reduce the time taken to remove the parts from the substrate. A number of methods to removing these lattice structures from the substrate were reported by different

collaborative authors and varied from using a knife/scraper through to diamond saw blades, and wire cutting [47] [48] [46]. The latter being the most time consuming as parts needed to be sent off site, however as the cutting process imparted least stress onto the built parts it was least likely to distort or break elements of the lattice structure during removal.

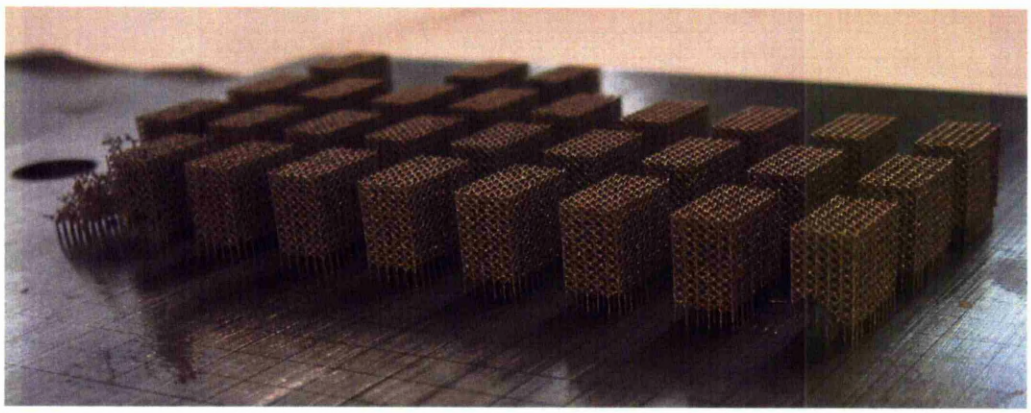


Figure 196: A set of lattice structures build with an array of processing parameters on top of an automatically generated support structure. Built in 316L steel. Lattice: with a laser power of 150W and a point exposure time of 1600 micro seconds. Support: laser power of 120W and a point exposure time of 900 micro seconds.

During the final stages of the research presented in this thesis an unexpected discovery was made after a warm up part was placed in an area of the build which was not covered by the substrate. The warm up part consisted of vertical pillars and was intended to allow time for the laser to stabilise before processing the test geometry. As can be seen in Figure 197 a proportion of the pillars were located off the substrate.

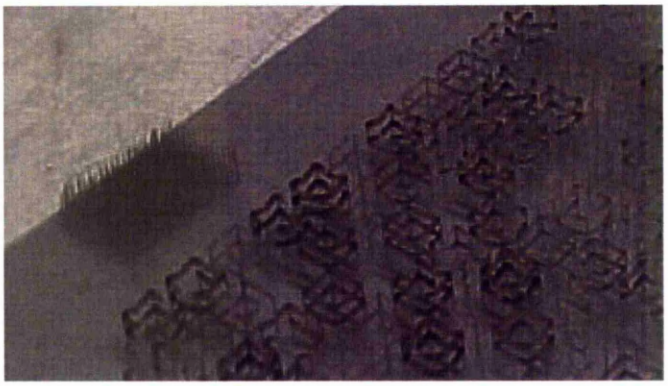


Figure 197: Image of a horizontal link test build showing a warm up set of vertical pillars.

During removal of the test build from the machine it was noticed that a number of the pillars not connected to the substrate had partly formed as seen in Figure 198. The lower section of the links had been disturbed by the wiper but not far enough from its original location for the subsequent layer's melt to fail to adhere to it. As this process repeated the structure seems to have gained stability as the amount of powder that was supporting the pillar increased. This increase in powder appears to have reduced the movement of the structure during the add layer process. After a proportion of time the amount of link under the top surface in the powder bed had become adequate to hold the pillar still during the add layer process and the links became vertical.

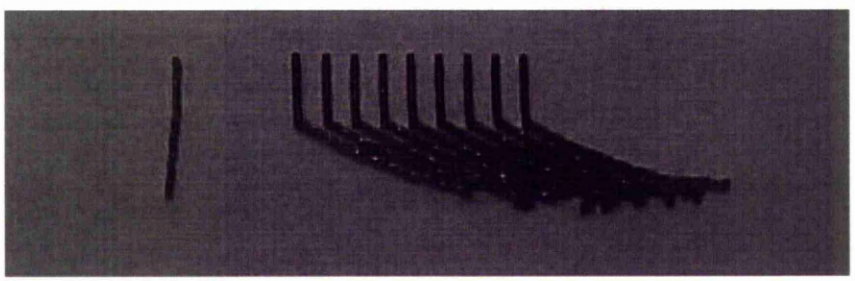


Figure 198: Vertical pillars that built in free powder off the edge of the substrate.

With this in mind a simple test part was created as shown in Figure 199 that made

use of the lattice structures previously discussed ability to recover from unsupported points of contact with the structure. Not all of the vertical pillars would have to build for there to be enough to build the cross-linking lattice structure before the final vertical pillars.

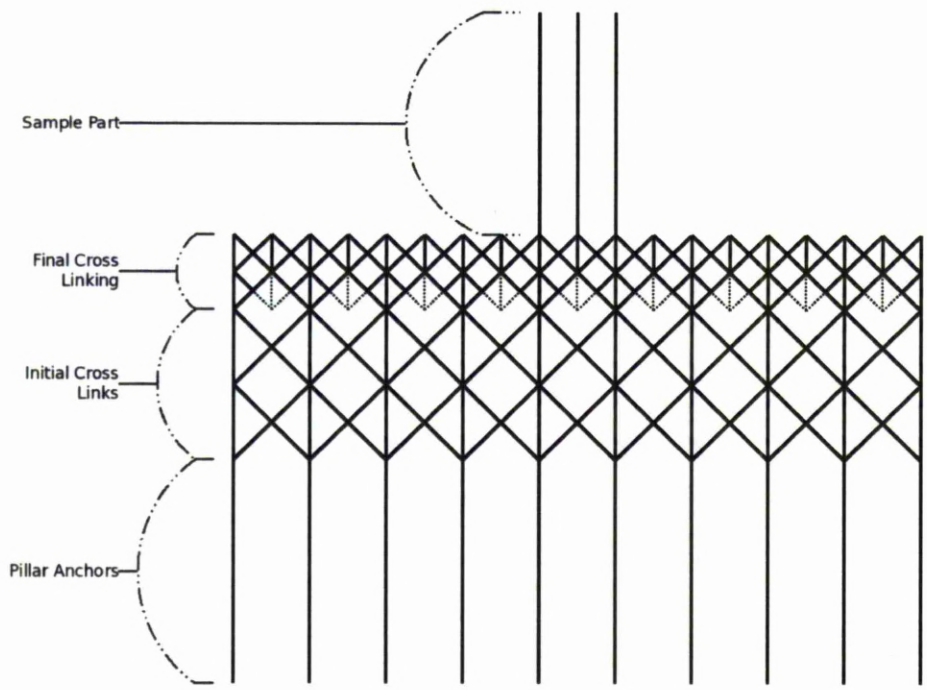


Figure 199: A diagram of a test part that makes use of the previously discussed ability of lattice structures to self support.

Figure 200 Shows a partial success from using this technique. Only a single test build was ran to investigate this. If developed further it could be useful in creating cores for composite structures but due to the stresses observed during building solid components it is assumed to be an unsuitable supporting method for lattice/solid mixed parts.

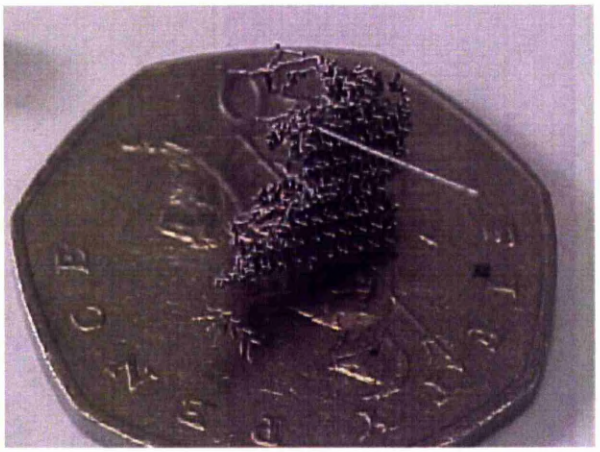


Figure 200: A proportion of the test part detailed in Figure 199 built in a free powder bed, 80W laser power and an exposure times of 1500µs.

6.2.2.6 Imported Lattice Geometries

As discussed in the earlier section of this thesis detailing the research and development of the software once it was possible to slice 3D lattice geometries it was also possible to import geometries from other sources. For clarity the figure shown earlier is repeated in Figure 201.

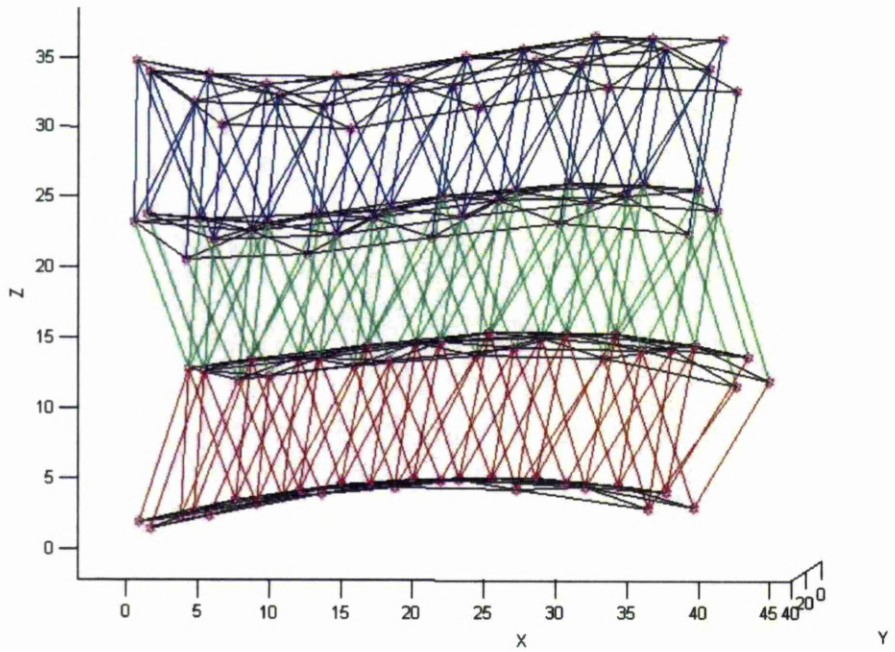


Figure 201: Visualisation of data being imported - repeated for clarity.

The results of importing this structure and processing with the same slicing techniques employed for the regular lattice structures is shown in Figure 202. Once again the need for further research into the processing of low angle links is highlighted here by the failure of many low angle links to process correctly. This build was ran for D. Rosen of The Georgia Institute of Technology and the results of this work were discussed during the presentation of a conference paper by C. Williams [40].

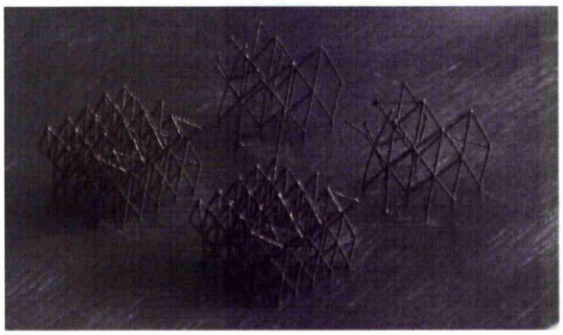


Figure 202: Resulting build from imported lattice data. Built in 316L steel with a laser power of 160W and a point exposure time of 2000µs.

6.2.3 Low Angle Links

Attempting to build lattice structures with elements lower than 30 degrees risks the elements not forming. This has also been reported by other collaborative authors; [18], [46], who made use of the features of the developed software that allowed the importing of lattice structures from external sources. Initial research showed that the lattice structures have a wide processing parameter window whilst working with the standard structures that contained links in the range of 90-45 degrees from the horizontal. Further research detailed in the following section focuses on investigating if greater success was possible by optimising processing parameters. Later research focuses on changing the scanning strategy employed to create the lower angle links.

Figure 203 shows a diagram detailing the test geometry utilised in testing low angle elements. This geometry allowed the simple and rapid evaluation of the effect of different processing parameters and scanning strategies on links built at specific angles. Each sample of the geometry had ten pairs of elements or nine pairs, and one single - if a vertical element was tested. Each pair comprised of a left and right hand element built at a specific test angle from the horizontal. These

test pairs where spaced from the substrate by a supporting structure built at different processing parameter sets to the elements in test. This supporting structure was dual purpose; to enable easy identification of a set of ten elements on a densely populated build, and to avoid misleading results caused by initial layer problems. The supporting structure was built with the same processing parameters on all test parts in each build. If a number of test parts failed to build the use of this supporting base meant it was possible to determine that this build should be repeated if parts failed to bond to the substrate. Without the supporting structure it would have been impossible to differentiate a structure that was not building due to its processing parameters, from one that failed to bond to the substrate due to problems with the initial layers, which may be the result of machine malfunction or poor set up.

An element was deemed successful if it formed, the primary purpose of a link in the structure being to connect two points. The number of successful elements per set of sample parts built with the same parameter sets is compared to others to determine the effect of the parameter or scan strategy change. Before the research contained in this thesis it was considered not possible to create elements much below 40 degrees, and so a major aim of the research was to get these elements to form.

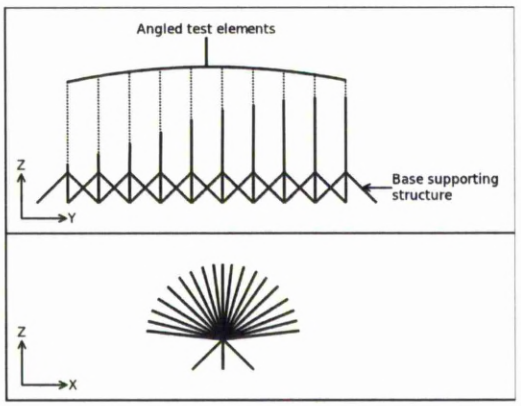


Figure 203: Diagram of low angle element test part.

6.2.3.1 Results of Varying Processing Parameters

The preliminary study developed an understanding of the relationship between the probability an element would build and its angle from the horizontal. The same processing parameters were used for all test elements on the first build. 25 samples were used so that the results gave an accurate representation of the probability of each link building at a particular angle from the X-Y plane.

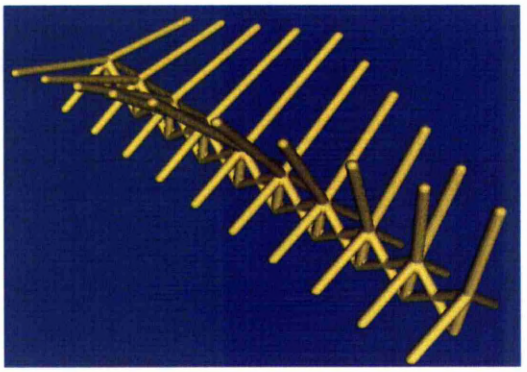


Figure 204: Computer rendering of link angle test part.

Figure 204 shows a test part that was created with the supporting structure and pairs of links at various angles. The angles of the links are detailed in Table 22.

Link Number	Link Angle (Degrees)
1	90.00
2	78.69
3	63.43
4	59.04
5	51.34
6	45.00
7	38.66
8	30.96
9	21.80
10	11.31

Table 22: Link angles of the initial set of low angle link trials.

At this stage there was also no understanding about how quickly the probability of the links forming reduced to zero. A second test varied the processing parameters to evaluate how much the success of the lowest angle links could be improved by altering the processing parameters.

Laser Power (mA Pump Current)	Laser Exposure Setting		
	2500µs	3000µs	3500µs
1500	1	3	6
2000	2	4	7
2500	Trial set	5	8

Table 23: Laser power and exposure settings for first low angle link builds built in 316L steel.

Both builds of the low angle links were built in an array of twenty-five parts. The first build - the trial set - was created using a safe laser power and exposure setting that had been used to create lattices with success before. The second build contained eight parameter sets as shown in Table 23, none of which were the same as the original trial set which is marked as “Trial set” on the table. All sample sets in the second build had three parts each, apart from the eighth which had four. Earlier work into the crush profiles of two lattice topologies showed that more variation occurred across one build than in one position in many different builds.

This large variation across one build could have been due to there being more variation in parts built with the parameter set on test compared to the parameters used in the sample set. If not that then the variation could be position dependent. To avoid problems with the position of builds as much as possible the samples were randomly allocated a position on the build.

Increasing the amount of energy delivered to each intersection point did increase the success of links with an angle of less than 45 degrees but not with those less than 20 degrees. Figure 205 shows one of the low angle link test builds.

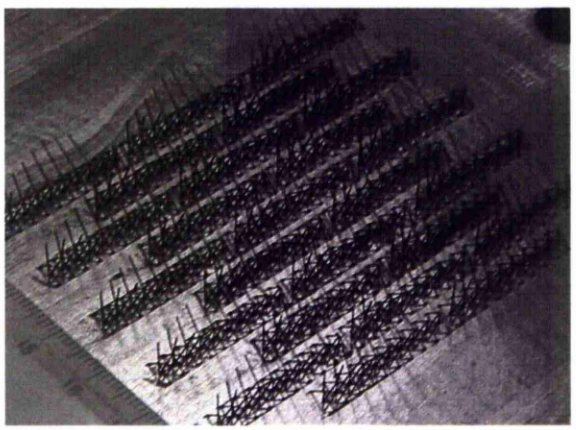


Figure 205: Test build 316L steel for 25 test parts with varying exposure time and laser power on intersection points as detailed in Table 23.

It is only possible to test discrete angles with single lattice elements. If the machines capability to produce thin walls was the intended investigation, a sample part could be designed where the angle to the horizontal of a surface would be gradually changed along its length. The starting angle would be selected as a known safe, reliable angle and would finish at, or slightly beyond an angle that is known to fail. The point at which the built test part begins to fail would determine a limit to the equipments capability to produce thin walls with a particular processing parameter set.

Table 24 shows the results from the first build of angled link test parts. All of the links on the 25 sample parts were successful over the inclusive range of angles from 90 to 51 degrees to the horizontal. The limit of the equipment's capability to produce fully formed links using the broad range of processing parameters (as shown in Table 23) is indicated by the complete success at 51 degrees, and incomplete success at 45 degrees. This suggests the finite limit of capability for the equipment lies between 51 and 45 degrees for the least successful parameter set in the build. None of the links at 11 degrees formed. This minimum angle, and the angle at which the structures begin to fail would be reduced with an optimised set of process parameters that cover a reduced range.

	Link Angle in Degrees									
	90	79	63	59	51	45	39	31	22	11
Fraction Formed	1.00	1.00	1.00	1.00	1.00	0.92	0.96	0.62	0.24	0.00

Table 24: Proportion of the links formed for all 25 parts in the initial low angle link tests as shown in Figure 205, using the processing parameters detailed in Table 23, built in 316L steel. A total of 25 links for 90 degrees, and 50 for the other test angles.

Within the range of laser power and exposure covered in this test it can be seen that raising the laser power has a greater effect on the success of the build than raising the exposure time. This is illustrated by the approximately horizontal line of Figure 206 and the upward gradient of Figure 207. This graph details the proportion of links that formed compared to their processing parameters.

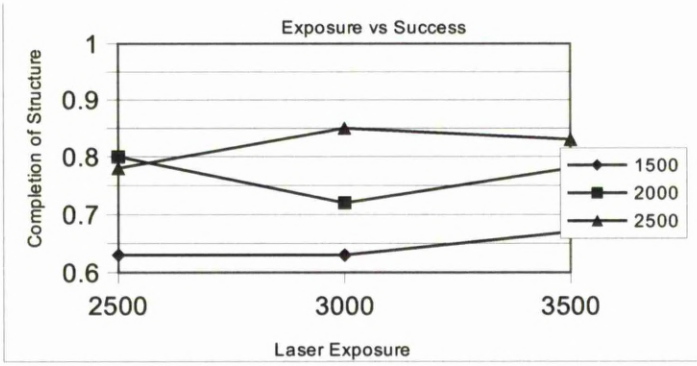


Figure 206: Success of links against exposure built in 316L steel. Legend shows laser pump current in as set in the material files fro the build ranging from 1500mA to 2500mA. Laser exposure time ranges from 2500µs to 3500µs.

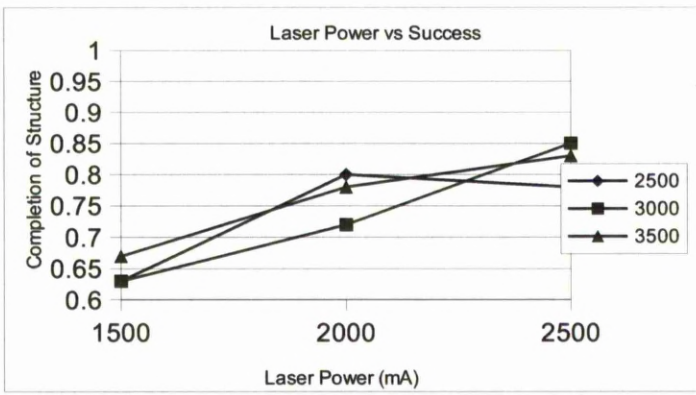


Figure 207: Success of links against laser power. Built in 316L steel. Legend shows laser pump current in as set in the material files fro the build ranging from 1500mA to 2500mA. Laser exposure time ranges from 2500µs to 3500µs.

This pattern of improving link success with increasing laser power was not expected as earlier work studying the link radius with varying laser exposure and power had shown increasing the exposure to have a more distinct effect on the truss radius than the laser power. A larger truss radius would imply a larger melt volume.

Figure 208 demonstrates that as the link angle gets closer to the horizontal the separation in the x-y plane of the slice and element intersections on consecutive

layers increases. For the same quantity of melt as the separation between the frozen melt pool on the previous layer and the new melt pool increases, the theoretical intersection between the new melt and the frozen melt reduces. Earlier work suggested larger laser powers created elements with a larger diameter. This would imply a larger melt, this would seem to suit the larger x-y plane separation encountered with low angle links more than a smaller melt volume. Increasing exposure time improving success was therefore unexpected.

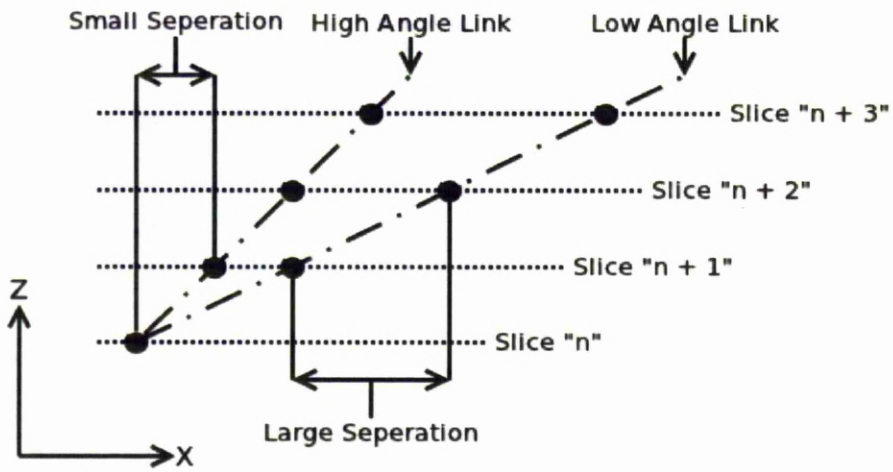


Figure 208: Increase in horizontal separation of intersection points on subsequent layers as link angle from horizontal reduces.

During these first two experiments trusses close to vertical on parts with the higher powers and exposures where observed to damage the soft wiper blade on the machine. While in some cases this may have not have caused the truss to fail, it did create wear in marks on the wiper. This is not desirable as it is preferable to have the ability to build structures at scales beyond the current SLM equipment limit of 200mm build height, and towards the 500mm build height which is currently under development. Figure 209 shows how overbuild can be tolerated up

to one layer thickness in height from the powder layer being processed. This is because the elevator has moved a layer thickness down before the wiper moves over the parts.

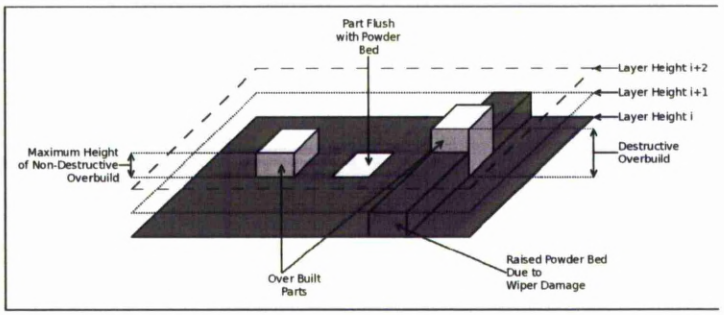


Figure 209: Demonstrating wiper damage caused by the parts overbuilding and rising above the process level.

If the elements cause significant wear to the wiper over this short build height, it is unlikely that a large lattice structure could be built on the new machine without one or more stops mid-process to change the wiper blade. Aside from the increase in build time and required operator intervention this would cause additional undesirable effects such as changes in oxygen levels.

6.2.3.2 Results from Scan Vectors

The trials investigating the creation of lattice elements using vectors rather than single exposure points used test geometries with elements within a 45.00 to 5.71 degree range as detailed in Table 25.

Link Number	Link Angle (Degrees)
1	45.00
2	41.99
3	38.66
4	34.99
5	30.96
6	26.57
7	21.80
8	16.70
9	11.31
10	5.71

Table 25: Link angles of the initial set of vector low angle link trials.

During the first build the effect of starting the scan from over previously melted material on the last layer and travelling along the vector out to a position over unprocessed powder was compared to scanning in the reverse direction, from scanning on a free powder bed back to the area over the previously melted material on the last layer. Figure 210 visualises a pair of test elements where the scan vectors belonging to the left hand element always starts over fresh powder, and the vectors belonging to the right hand vector always starts over a solid element.

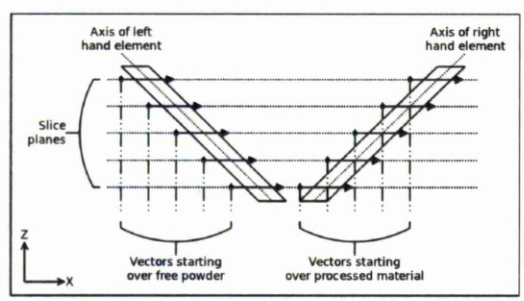


Figure 210: Diagram of scanning from over powder, and from over previously processed material.

The effect of this is clear in Table 26. In each case the fraction that are successful

is less for completed links scanning from over powder to over solid (left) to those starting over solid and scanning out to over powder (right).

This could be explained by the results of single scan track tests carried out by Childs et al [27] which were shorter than the distance scanned. In Childs et el's work the laser passed once over the powder along a single line and therefore the powder was heated by the laser and formed a melt which subsequently rapidly cooled to form a metal track. The start of the frozen metal track did not start in the same place as the laser started processing, the difference being greater than could be explained by shrinkage alone, and suggested a tendency for the melt to initially follow the laser as it tracks across the bed. Applying this phenomena to the lattice elements could help explain the difference in the number of lines fully formed.

While scanning the lines from over powder towards the previously processed material the system is scanning on a free powder bed. During this section the start of the frozen melt pool would advance along the length of the scan vector as it is free to move until it freezes. This would result in a greater distance between the start point of the vector on the following layer until the point where it is melting directly over previously melted material. When scanned the other way the previously melted material provides a substrate to which the new melt would wet. The wetting onto the this solid surface acts to stabilise the new scan track by reduce the tendency to ball up, as described by the Rayleigh instabilities and the contact with the existing structure increases the rate of cooling and freezes the start of the link quicker. Since the start point is wetted and and anchored to the structure created during the previous layers it is not free to move along the length of the vector and create contraction greater than can be explained by thermal

shrinkage alone.

Material File	Left		Right		Total	
	Number	Fraction	Number	Fraction	Number	Fraction
0	1	0.01	22	0.28	23	0.14
1	8	0.10	34	0.43	42	0.26
2	30	0.38	46	0.58	76	0.48
3	35	0.44	63	0.79	98	0.61
4	47	0.59	61	0.76	108	0.68

Table 26: Proportion of formed links for initial low angle link test built in 316L steel. Processing parameters detailed in Table 27

Material File	Exposure (µs)	Laser Pump Current (mA)	Vector Size (mm)
0	400	1000	0.5
1	400	1250	0.5
2	400	1500	0.5
3	400	1750	0.5
4	400	2000	0.5

Table 27: Processing parameters used for initial trials of replacing single exposure points with vectors.

Table 28 Shows the results of a similar experiment where the scan vectors went from over solid to over free powder. This shows that the pattern observed in Table 26 was not due to machine related issues such as the effect of gas flow over the process chamber as the difference between the success of left and right hand vectors was far less for a given process parameter set.

Vector Size	Left		Right		Total	
	Number	Fraction	Number	Fraction	Number	Fraction
0.5	57	0.71	59	0.74	116	0.73
1.0	48	0.60	66	0.83	114	0.71
1.5	71	0.89	72	0.90	143	0.89
2.0	71	0.89	74	0.93	145	0.91
0.0	29	0.36	28	0.35	57	0.36

Table 28: Fraction of successful links for second vector low angle link test. Processing parameters were constant at 2500µs exposure, and 2000mA pump current. Test angles are detailed in Table 29.

The experimental results shown in Table 28 show the effect of varying the length of the scan vector used. The control group vector size is 0.0mm, and so scanning with points rather than vectors. This clearly demonstrates the increase in the number of fully formed links when spots are replaced by vectors. The worst total fraction of successful links from the parts built with vectors was 0.73 with a range from 0.73 to 0.91, whereas the fraction for the links processed with spots was 0.36.

The test geometry used to collect the results shown in Table 28 had links at a lower range of angles than previously attempted. This explains the lower level of success with the control sample built with spots when compared to the results from structures built with vectors. The angles used in the samples for this build are detailed in Table 29. Take note of the lowest angle of 2.9 degrees.

Link Number	Link Angle (degrees)
1	26.57
2	24.23
3	21.80
4	19.29
5	16.70
6	14.04
7	11.31
8	8.53
9	5.71
10	2.86

Table 29: Link angles for initial low angle link test.

Figure 211 shows a full test build after removal from the machine. An adverse effect of using vectors is that if the angle of the link is to great, i.e. too close to the vertical then the links appear like thin wall structures, rather than cylindrical trusses.

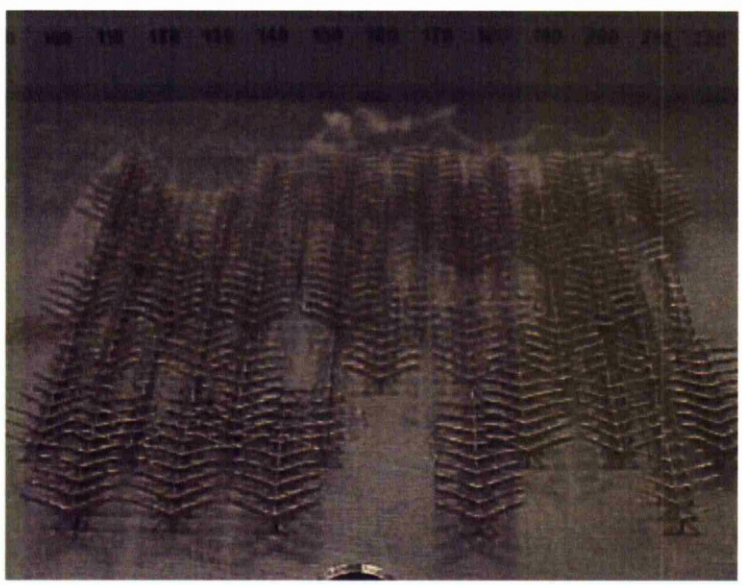


Figure 211: Image of full low angle test build in 316L steel showing parts built with vectors. Processing parameters were constant at 2500µs exposure, and 2000mA pump current. Test angles are detailed in Table 29.

Figure 212 explains why this is happening. When a cylindrical object is sliced the resulting section gets progressively closer to being circular as the cylinder

approaches being perpendicular to the slicing plane. Therefore the area the laser scans should approach a 1:1 ratio of width to length in order to approximate the perfect intersection area. When scanning with vectors the width of the scan track is the minimum thickness the experimental equipment is capable of at the laser parameters used, and its length is the scan vector length. This results in an element with a rectangular, rather than square, or circular section.

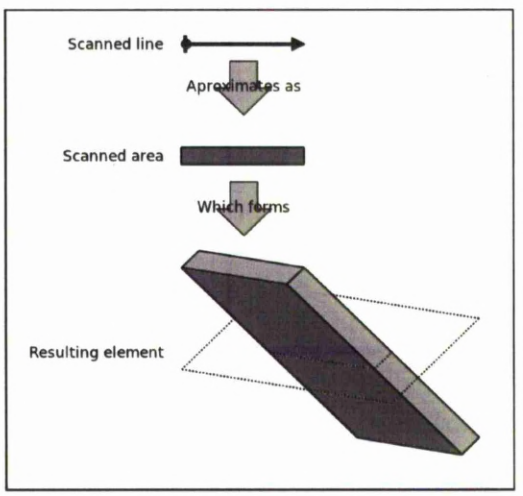


Figure 212: Detailing how the use of scan vectors on elements close to vertical creates a thin wall like element.

Figure 213 shows a test part with links that where fully formed at an angle of 2.9 degrees from the horizontal. The links have fully formed but have a rough, uneven appearance. The links appear much rougher than those built at between 90 and 45 degrees where the intersections are processed with single spot exposures.

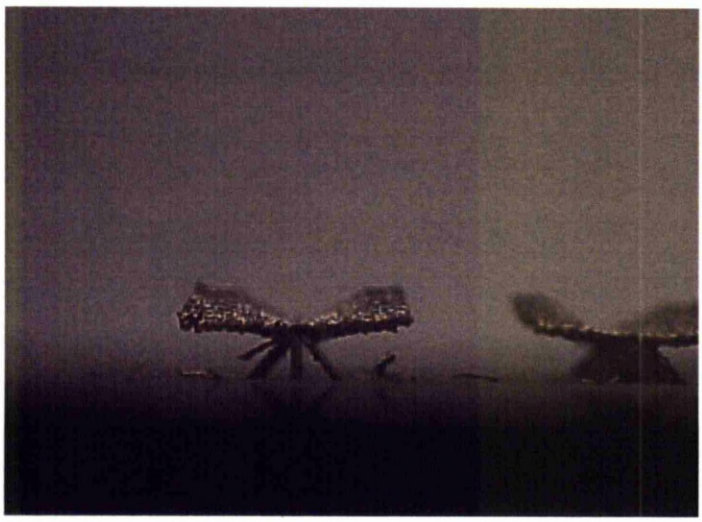


Figure 213: Close up image of a truss built at 2.9 degrees. Vector size was 2mm, processing parameters were 2500µs exposure and 2000mA pump current.

In situations such as detailed in Figure 214 the cylinder that represents the intended geometry of the lattice element is approaching becoming parallel to the slicing plane and intersection area between the cylinder and the slice plane is a high aspect ratio cylinder. Figure 214 shows why the long scan vector is a better approximation to the slice intersection in this situation than in the near vertical element. In this situation the scan line is a close match to the high aspect ratio ellipse, and so the resulting element is closer to being cylindrical than before. This, and the previous results that demonstrated the interplay between the proportion of elements formed and changes in the processing parameters indicate that any final end-user application would have to automatically adjust parameters up to the critical element angle, then start to use scan vectors of increasing length.

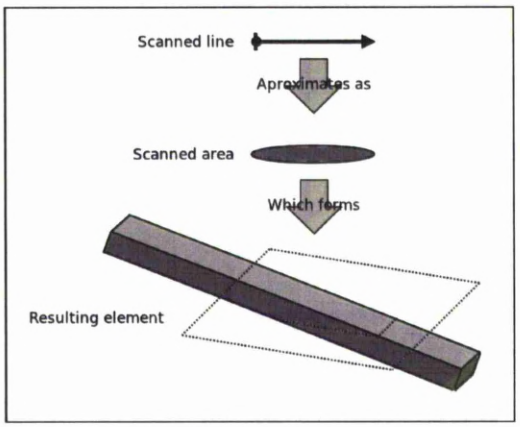


Figure 214: The use of scan vectors on elements close to horizontal creates a link closer to cylindrical.

6.2.4 Horizontal Links

As discussed in the code development section of this thesis six scanning strategies have been developed to generate slice data to run the experimental equipment required to create lattice structures that include horizontal links. For clarity the test sample as shown before is repeated in Figure 215.

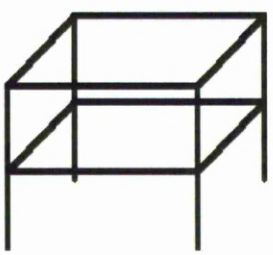


Figure 215: Horizontal link test part.

The build arrays for these experiments were automatically populated using specifically developed software that was also used for the previous low angle test parts. This software placed test samples randomly into a defined array. Due to this, and averaging over a number of parts, any positional related errors are reduced.

6.2.4.1 Horizontal Links Single-Pass

Scan Strategy 1 (Figure 216) melts the first point on the line followed by the last then continues to target the laser at points on the line on progressively closer to, and on alternate sides of the centre.

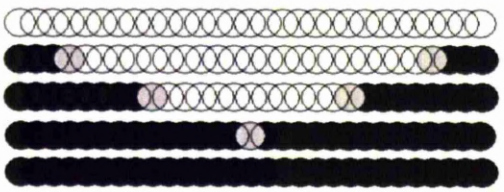


Figure 216: Scan Strategy 1 – figure repeated for clarity

The first scan strategy showed less than one percent complete links. The complete test build can be seen in Figure 217.

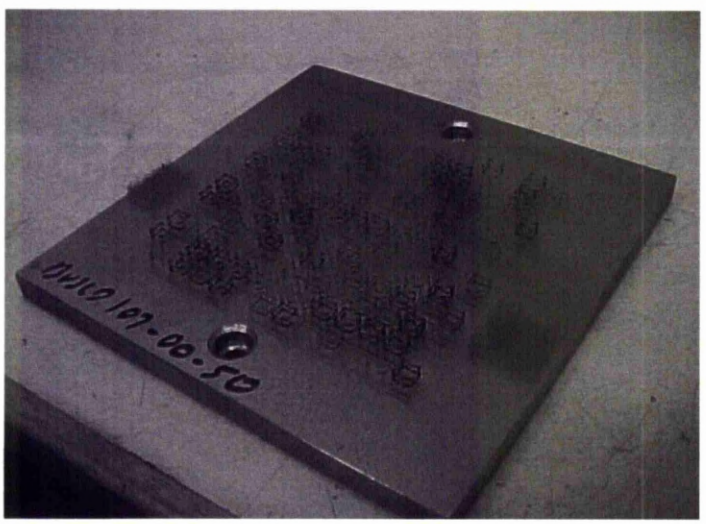


Figure 217: Scan Strategy 1 build. Built in 316l steel using process parameters detailed in Table 30.

		Power W				
		40	80	120	160	200
Exposure µs	200	1	2	3	4	5
	400	6	7	8	9	10
	600	11	12	13	14	15
	800	16	17	18	19	20
	1000	21	22	23	24	25

Table 30: Process parameters used in horizontal test builds.

Some links were nearly complete with a small gap in the centre of the link as shown in Figure .

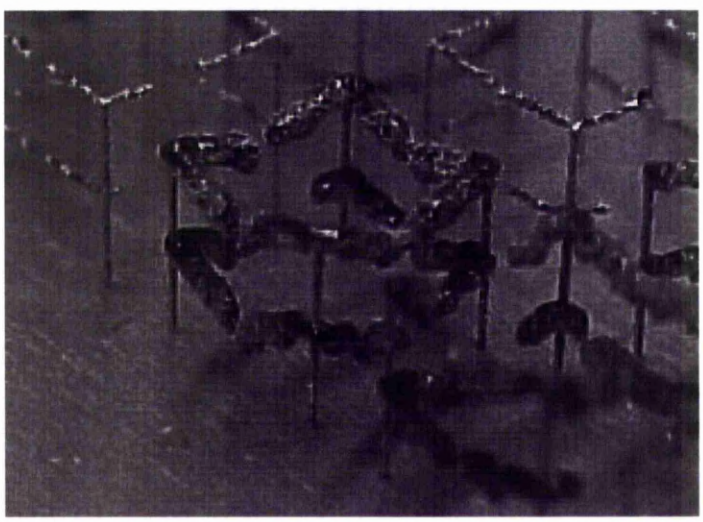


Figure 218: Scan Strategy 1 part close-up. Part built in 316l steel using 1000µs exposure and 80W laser power.

There was also a wide variability in the appearance of the sections of the links that did build. Horizontal links that are a similar diameter to the vertical pillars are considered to be the most desired result and will be referred to as of 'good appearance', where the overbuilt and drooping links are referred to as 'drooping'.

Figure 219 details the drooping phenomena observed where the horizontal link formed drops into the powder bed in the centre, where it has thicker section than close to the vertical pillar elements.

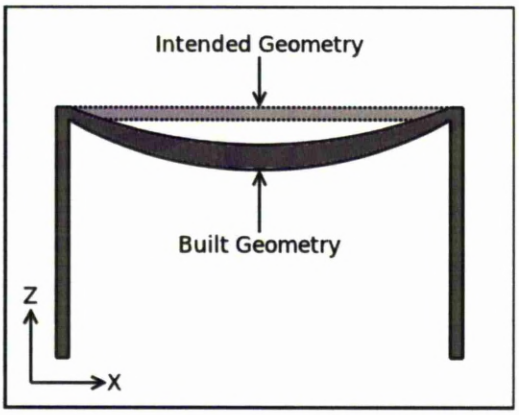


Figure 219: Diagram of a drooping element.

Figure 220 shows an over built link where the width of the element is far thicker than the intended geometry and does not approximate to the vertical elements of the test geometry.

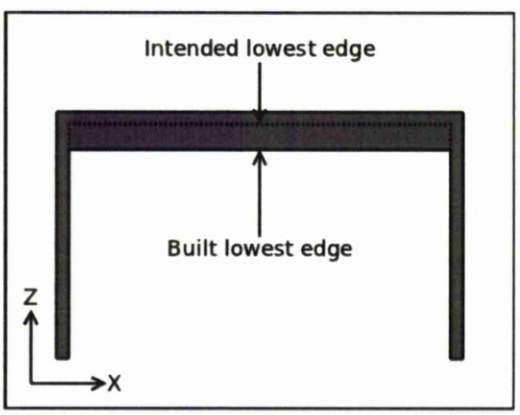


Figure 220: Diagram of an overbuilt, thick element.

Scan Strategy 2 (Figure 221) targets the laser at the first point on the line, the point closest to a 1/3 of the distance along the line, and a final point 2/3 of the distance along the line. This process then repeats with the following points being the second point in the line, point closest 1/3 plus 1, and the point closest 2/3 plus 1. This sequence repeats until the laser has been targeted at all the points on the line once.

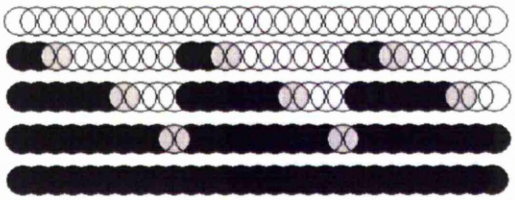


Figure 221: Scan Strategy 2 – figure repeated for clarity

Figure 222 shows the result of processing links using Scan Strategy 2. Sections of the elements frequently sprung up out of the powder bed while being processed by the laser. In situations where these sections had not completely come away from the rest of the test geometry they were often destroyed by the subsequent wiper motion. These collisions between the sections of the lattice and the wiper also frequently removed the supporting vertical structures from the substrate. The parts shown in Figure 222 are at best missing the whole centre section of the three segment scan. From this, it is assumed that starting a section of a link on free power was not a suitable approach to take.

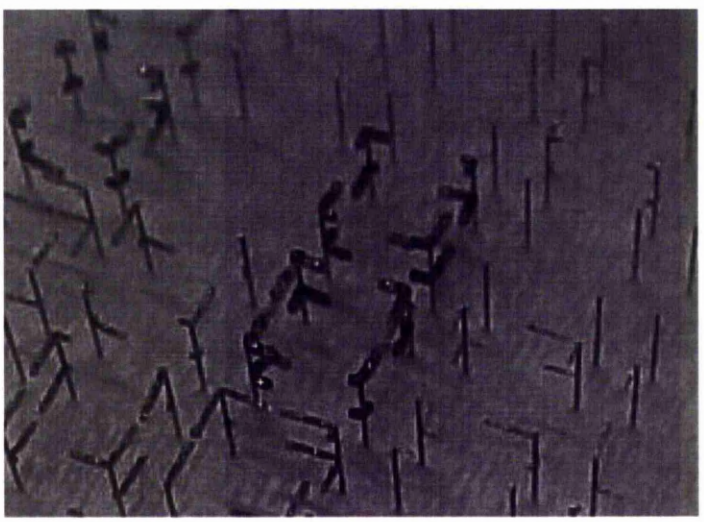


Figure 222: Scan Strategy 2 part close-up. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

An aim of this section of the research was to find out if there are scanning strategies that can be employed to enable the formation of horizontal elements of a

similar diameter and morphology to the vertical elements. The experiments detailed previously created occasional complete links that spread between two points, traversing a free powder bed parallel to the slice plane. These links were however far from perfect. The methods used as discussed thus far suffered some critical defects that limited their usefulness. These defects were; poor level of success, much bulkier links than the vertical elements, often drooping, and also often colliding with the wiper causing damage to the rest of the structure. As the bulky but complete elements have little practical use stopping the formation of the gap between the finer horizontal elements was critical. If these deviations from the intended geometry can be removed or reduced then the use of low angle and horizontal elements in lattice structures would be a realistic proposition.

The formation of the gap in the incompletely formed links may be related to the additional melt phenomena observed by Childs et al 2005 [27], where they observed the two melted regions within the powder bed, one in front and one behind the laser beam as the laser scanned across the powder bed. The alternate point melting pattern of Strategy 1 might produce a similar problem as the points move closer together. When one point on the line is processed it could be melting part of the line on the opposing half of the link. This could lead to a increased volume of melt which begins to ball, pulling the free end of the link back toward the end that is welded to the vertical pillar element. A second possible cause is that as the points get closer together at the centre of the line the time the laser spends jumping between the points decreases and so the delay between the points being heated reduces and so as the points approach the temperature of the melt pool increases, and it is possible that two melts pool coexist robbing powder from

around and between them. Evidence that this is happening may be seen in Figure 218 where the diameter of the elements on the part to the top right of the figure increase towards the centre of the link. A third possibility is that the temperature when the points are close, may be high enough to eject molten melt from the centre of the link.

The second strategy suffered from sections of solidified material being ejected from the part bed, the ejection occurring toward the end of the building of the link. As the laser beam processes a link on the powder bed a proportion of the molten material is thrown out producing a small shower of sparks. The ejection of this material appears to displace the section, with the material being ejected lifting the solidified section, or the recoil on the powder bed of these ejections pushing up one end of a segment lifting it out of the bed.

6.2.4.2 Horizontal Links Multi-Pass

Scan Strategy 3 (Figure 223) start with Strategy 1 followed by an additional pass of the laser. This pass starts from the point closest to ¾ of the distance along the line and remelts the sequence of points in order to ¼ of the distance along the line. The intention of this was to bridge the gap by pulling a melt over the gap and trying to encourage the melt to wet to both sides of the horizontal links formed with the initial pass of the laser.

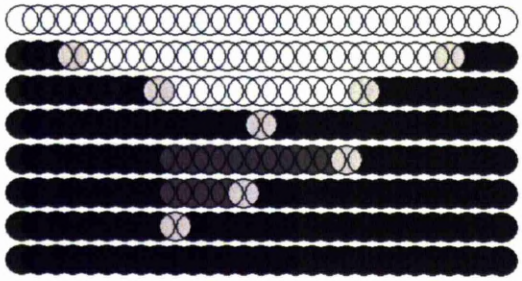


Figure 223: Scan Strategy 3 – figure repeated for clarity

The parts built using the third strategy (Figure 223) showed significant balling of the melt pool. On occasion, these balls on the links would be hit by the wiper pulling the part from the substrate. This strategy was the first of the multi-pass strategies, with the second pass of the laser occurring immediately after the first pass.

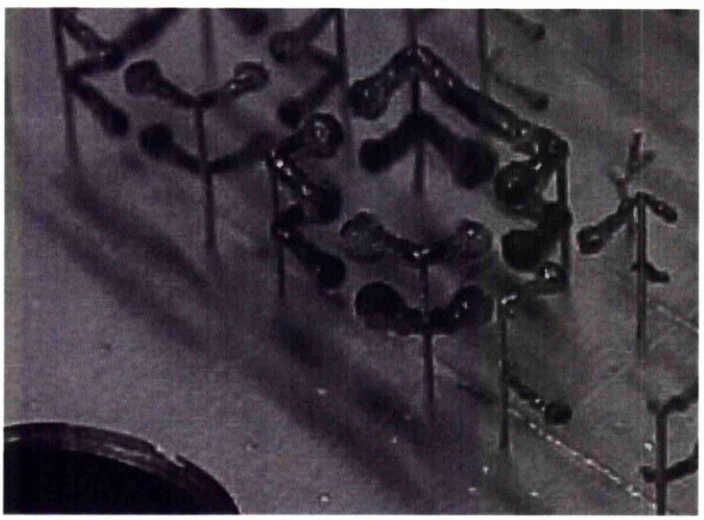


Figure 224: Scan Strategy 3 part close-up. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

A possible reason for the balling was that too much energy was delivered in too short a time into the links. This would reduce the viscosity of the melt, increase the size of the melt pool and cause larger temperature gradients that would increase the Marangoni forces. The final two scan strategies were adjusted so that

the first pass of the laser was made for all four links, before the second pass on each link. This gave the links time to cool between passes which was intended to reduce the volume of melt by giving it time to cool and solidify.

Scan Strategy 4 (Figure 225) also starts with Strategy 1 followed by an additional pass of the laser. The second pass of the laser for scan strategy 4 was the opposite to the first pass, the laser starting in the centre of the link and working out to the edges.

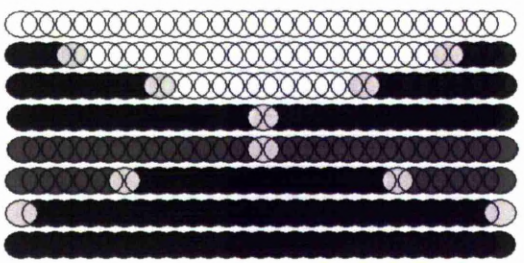


Figure 225: Scan Strategy 4 – figure repeated for clarity

Strategy 4 showed a higher proportion of links that connected the opposing vertical pillars than observed with the previous three strategies. Figure 226 shows the appearance of the build before it was removed from the powder inside the process chamber. The effects of the wide range of processing conditions can be seen with some links pulling in the surrounding powder leaving the link in a valley.

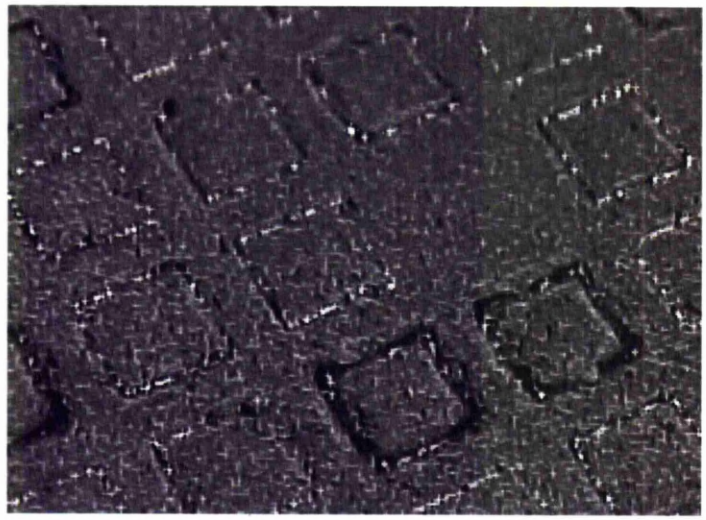


Figure 226: Scan Strategy 4 build in powder. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

Figure 227 shows the same test as shown in Figure 226 after removal from the machine and powder.

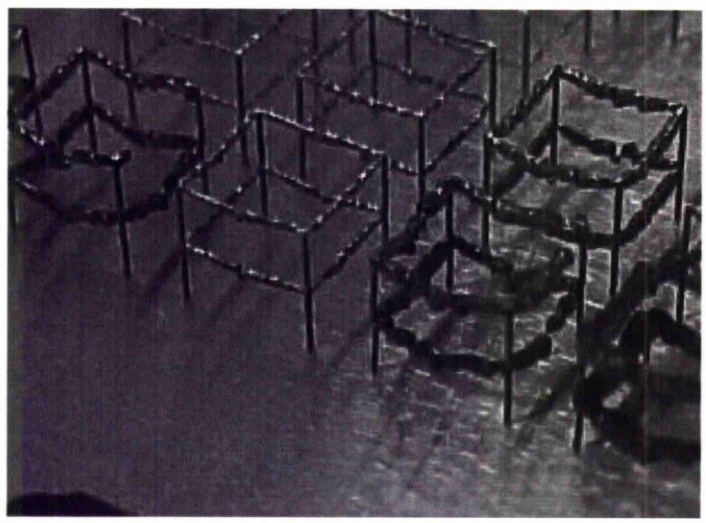


Figure 227: Scan Strategy 4 part close-up. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

The robbing of powder often occurred with links that also drooped from horizontal. It appears that powder surrounding the link is pulled into the melt from in front and beside the focus point of the laser as the element was formed. This led to the link tunnelling into the powder bed. Many of the complete links had a good

appearance as described earlier, but most of the continuous links were much larger diameter than that of the vertical links. An alternative explanation could be that material was being ejected from the melt with enough force to blow powder away from the area surrounding the melt. If this ejection of powder surrounding the link was occurring on both sides and in front of the elements it could also have caused the tunnelling.

Scan Strategy 5 (Figure 228) is one pass of the laser covering the points in order from the first point on the line to the last followed by a reverse pass of the laser from the last point on the line to the first. The intention of this strategy was to create a melt that wetted and anchored to the vertical pillar, then moved across the bed. The second stage of the scan wetted and anchored to the opposing vertical pillar and pulled the melt in the opposing direction back over the link segment created on the first pass, connecting the link segments.

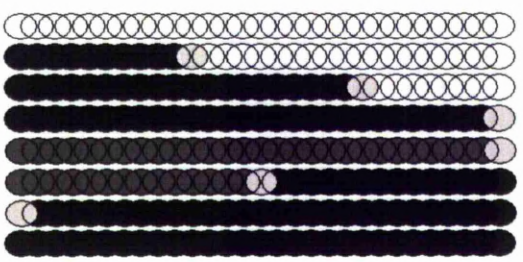


Figure 228: Scan Strategy 5 – figure repeated for clarity

Figure 229: Scan Strategy 5 build in powder. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

Although the final strategy tested had the same delay in between the first and second scans of the laser it suffered from similar problems to the third scan strategy with excess balling apparent as shown in Figures 229 and 230. The could have been exacerbated by the order of the scan points meaning adjacent points were exposed consequently resulting in very little delay between exposure, and thus creating a larger melt pool volume.

Figure 230: Scan Strategy 5 part close-up. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

Figure 230 shows the results of the test shown in Figure 229 after the test was removed from the machine.

		Power W				
		40	80	120	160	200
Exposure µs	200	0	1	4	3	5
	400	0	4	6	5	6
	600	2	6	6	6	6
	800	4	6	6	6	6
	1000	4	6	6	6	6

Table 31: Details number of test parts that had links that were close to successful built in 316l steel using scan strategy 4. Light grey highlights parameter sets with 5 out of 6, dark grey highlights 6 out of 6.

Table 31 shows the number of parts which had at least one nearly complete link on each of the four sides of the part. This shows that increasing the laser power and exposure tended to give a higher quality of part. However this gives an over simplified view as the parts with the highest laser power and exposure are also more susceptible to drooping, as shown in Table 32.

		Power W				
		40	80	120	160	200
Exposure µs	200	0	0	0	0	0
	400	0	0	0	0	0
	600	0	0	1	6	6
	800	0	0	3	6	6
	1000	0	0	6	6	6

Table 32: Details number of test parts that had drooping links built in 316l steel using scan strategy 4. Dark grey highlights 6 out of 6.

This region of the process settings chart frequently gave rise to drooping links. The information shown in Table 33 identifies the parts which were both nearly successful and also of good appearance.

		Power W				
		40	80	120	160	200
Exposure µs	200	0	1	4	2	1
	400	0	4	5	0	0
	600	1	6	2	0	0
	800	2	5	0	0	0
	1000	4	2	0	0	0

Table 33: Details number of test parts that had links of good appearance built in 316l steel using scan strategy 4. Light grey highlights parameter sets with 5 out of 6, dark grey highlights 6 out of 6.

Table 34 Shows the fraction of horizontal links, built using each set of processing parameters, that were continuous from one vertical pillar to another. As there are six parts per set, there are 24 horizontal links for each parameter set. This shows that it is possible to reliably build horizontal links, however these links appear solely in the region of the processing window that also suffer drooping links as shown in Table 32. This means the links are substandard when compared to the vertical elements of the parts. The horizontal links appear over built and not conformal to the intended geometry.

The horizontal elements built using this technique would perform differently in mechanical loading because of the difference in size and conformance to intended geometry when compared to the elements built vertically. Predicting the behaviour of the bulk lattice structure requires consideration for these differences, and would have to be taken into account when predicting the tensile properties. This implies generating computer models that would enable prediction of bulk material properties based on a lattice topology would need to be far more complex. These models would be very challenging to validate.

		Power W				
		40	80	120	160	200
Exposure µs	200	0.00	0.00	0.29	0.29	0.58
	400	0.00	0.42	0.67	0.50	0.67
	600	0.00	0.56	0.92	0.83	0.88
	800	0.13	0.56	0.98	1.00	0.90
	1000	0.17	0.85	1.00	0.90	0.94

Table 34: Details fraction of complete links built built with particular processing parameters in 316l steel using scan strategy 4. Light grey highlights parameter sets with over 0.9, dark grey highlights parameter sets that created parts with all links.

The excess balling with the first double pass scanning strategy (Strategy 3) could have been due to the melt pool reaching too high a temperature and this may increase the size of the melt pool as the surrounding powder also melts. There is also an asymmetry between the balls created on each side of the link, as the second pass of the laser beam moves from one side of the link producing a temperature gradient and thus a flow of material. This differs from the symmetrical first and second passes of the laser used in Strategy 4.

As it was not possible to use different laser parameters for the first and second

laser passes due to the limitations of the experimental equipment the only way to lower the melt temperature during the second pass was to introduce a delay between the first and second pass. This reduced melt pool balling for the fourth scan strategy, but not significantly for the fifth scan strategy. With the fourth scan strategy, the first pass of the laser had two converging heat fronts, starting at each end of the horizontal link and moving towards the centre, the laser focusing at a point on one half of the link before jumping over to focus on a point on the other side of the link. These jumps between the melting points leads to a large - although not perceivable to the naked eye - delay between melting consecutive points which may allow the average melt temperature to drop. The fifth scan strategy did not have these jumps and thus the melt had less time to cool, which left it more susceptible to balling when processed with the same laser settings.

These initial studies on the horizontal links demonstrated that creating the links with a single pass of the laser was not successful. This is supported by the Rayleigh Instability theory that suggests cylindrical pools of liquid will break up if the ratio of the length of the liquid cylinder divided by it's diameter is greater than π, in order to minimise surface energy. Childs et al demonstrated a link between this theory and the behaviour of scan tracks on a free powder bed. Links processed with multiple passes of the laser where more successful.

Although some of the links formed fully the results where not as desired. The links where much larger, and often dipping in the centre of the element below the intended level of the link. The most successful scanning techniques made use of delays between successive exposures of the element. One type of delay was created by jumping the laser from one side of the link to the other. This added a

delay with the laser off as the scanners moved the target of the laser from a point on one side of the element, to the next exposure point on the other half. A secondary delay was added by completing the first pass of all four links before returning to complete the second pass on the first and following three.

6.2.4.3 Results From Pulsed Scanning

The five scan strategies attempted using the standard control software demonstrated the critical requirement to control the melt pool. In order to progress the research further bespoke software was developed that replaced the front end control software on the experimental equipment. The development of this was discussed in detail in the previous section. This software was responsible for feeding the scanners with slice data as required at the start of each slice. This removed limitations in the number of available parameter sets, and critically allowed the addition of a pause between exposure points.

Due to the way the pause was introduced the delay time could not be finely controlled and was approximately about 0.75 seconds. During testing it was observed that the delay time was great enough for the melt to solidify and cease emitting light in the visual spectrum.

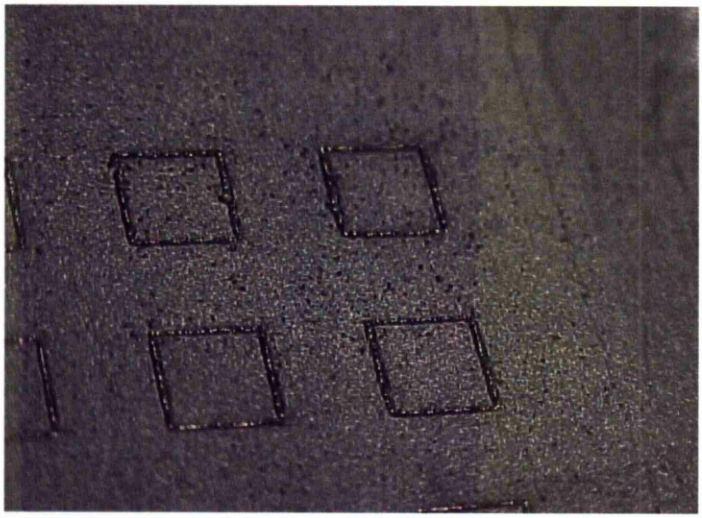


Figure 231: View of trenching of horizontal links on powder bed surface. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

During the experimentation a trenching phenomenon was observed. When the link is viewed directly after processing (before another layer of powder is added over the top of the links) it appeared to be inside a trench on the powder bed surface. There was a small gap running along the length either side of the links and in some examples the link appears to be deeper into the powder bed the further from the starting point of the link. This can be seen as shadows either side of the top two links in Figure 231. The likely hood of trenching occurring was greater with increasing power or exposure time, and decreasing the distance between the exposure spots. It is assumed that increasing the laser power increases the quantity of vapour/metallic gas created. This could have blown powder away from close to the melt pool that has not been processed. Decreasing the distance between exposures increases the number of exposures, and could also have increased the quantity of vapour/metallic gas that is created.

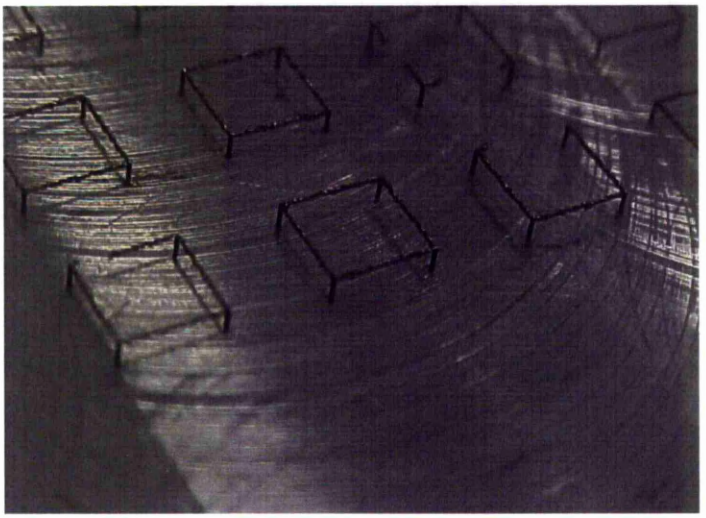


Figure 232: Image of successful horizontal links. Parts built in 316l steel over a range of process parameters as detailed in Table 30.

Figure 232 shows that the created horizontal links are much closer to size and appearance of the vertical links in comparison to those which where created in previous tests. Links created in the earlier test did droop, not staying as level as those shown above, and often suffered defects in the centre of the links.

It is preferable to aim for the simplest and most robust solution. It is expected that the earlier techniques would have benefited from separate processing parameters for the first and following passes of the laser, but this is a far more complex solution as the ideal process parameters for the second pass will be heavily dependent on the results of the first pass of the laser. In addition to this the earlier multiple pass scan strategies have a larger volume of metal in a liquid state at any stage, and so is naturally a more complex situation to deal with as the Rayleigh instabilities and Marangoni convection have to be considered, and possibly managed.

The pulse and delay method simplifies the creation of the horizontal links by creating them with a single pass of the laser. With this method there is only the

delay and a single set of processing parameters to optimise. A second pass of the laser over the link would introduce a second set of processing parameters. This second set of parameters is likely to require separate optimisation as the interaction between the laser beam and material is very different. This is because the second pass of the laser is is re-exposing a frozen melt, rather than a powder bed. Due to this this the proportion of the laser energy being reflected, the rate of absorption of laser energy into the material, and the conduction and radiation of heat from the re-exposed material are all likely to be different.

7 Conclusions and Recommendations for Further Work

7.1 Findings

The developed system was demonstrated as capable of producing lattice structures in the 5mm through to 0.5mm cell size range with elements in the region of 100-250 microns. This was made possible by controlling the creation of slice files and using an alternative raw geometry to the standard surface mesh to describe the lattice model.

The initial limit to low angle links in the structure was found to be between 50 and 40 degrees. Early investigations found that it was possible to create lower links by merely adjusting the processing parameters. Altering the scanning technique enabled the creation of links just below 3 degrees from the horizontal.

Many techniques where investigated in to create horizontal links in one layer. Successful links where created both when width of the link was very large, and the volume of melt was controlled by introducing delays in between exposures on the link. The latter gave preferential results as the links were much more similar to links produced between vertical and 45 degrees using the initially developed technique. The former produced links which were much wider than links produced in the vertical to 45 degree range.

7.2 Contributions of the Work

This research has shown that through precise control of the slice data an increased level of control over the laser than normally seen in LM equipment it has been possible to create lattice structures with a wider range of topologies, and cell sizes

some of which have not been demonstrated using other techniques.

The success of the low angle links and surfaces are also in contrast to the commonly understood limits of the techniques processing metal powders via SLM. These structures have links at far lower than the normal quoted limits which range from 30 to 45 degrees.

The developed software has enabled a significant quantity of further research into aerospace, heat exchange, and biomedical research. The tools developed during the course of this research have enabled users of the SLM equipment to develop lattice structures at a level of complexity previously unachievable.

7.2.3 Published Papers Which Made Use of Developed Techniques

Santorinaios et al [47] published a paper detailing the crush performance of lattice structures in various cell sizes and concluded that the unoptimised cell topologies performed well in comparison to currently available foams. The author also highlights the need for more developed design tools. It is worth noting that this research took place before the lattice generation tools were developed to include the custom cell geometries. The contribution to this work was experimental planning advice following the crush tests described earlier in this thesis and creating the files required to build the test geometries on the SLM equipment.

McKown el al [49] discussed the performance of structures created using the developed software techniques in blast and dynamic loading situations. The author identified different failure mode in structures with vertical pillars to those without. Without the pillars the collapse of the structures was observed to be progressive, while with the structures exhibited a diagonal failure plane after the vertical pillars

buckled. The article also discusses tailoring the structure to give designed failure methods and proposes the use structures in vibration control, load bearing and impact protection applications. The author of this thesis aided in the preparation of the files required to create the test geometries using the developed programs.

Tsopanos et al [18] investigated the effect of processing parameters on the mechanical properties of lattice structures created using the stainless steel structures. This investigated the bulk lattice structures as well as single elements produced large enough to test in low load tensile testing equipment. This research made use of the import of pre-defined 3D element geometries to produce the large vertical and angled strands that where not connected to anything other than the substrate at the base of the build. This further expands on Mckown's [49] observation that the structures could be tailored for specific applications that this could also include adjustment of processing parameters. Critically the author observes the variation in mechanical properties for a specific process parameter set is suitable to progress the research further into the use of the structures as cores in twin skin lattice/composite sandwich structures. Tsopanos used an instance of the software to create the standard lattice structures while the author of this thesis prepared files for the single strand tests and sliced the single strands in the developed programs.

Hasan et al [50] studied the differences in drop weight impact performance of titanium alloy lattice core structures and that of of aluminium honeycomb. The study shows the performance of the structures to be comparable. The impact damage was reported to be more localised the titanium alloy lattice structure which is a potential benefit highlighted for further research. The lattice structures

created for this work was larger than the previously mentioned papers. The samples were 100x100x20mm with 2.5mm cell size. This gave over 100 thousand elements. The authors of this paper were able to prepare test files with a copy of the software making use of the significant development effort (not discussed in this thesis) that had gone into making the software accessible to those without a programming knowledge.

Mullen et al [48] [46] detailed work towards using titanium lattice structures for orthopaedic implants with bone ingrowth and solid geometries. This made use of the software's ability to create very small cell size (0.6-1.4mm) lattice structures that can be shaped to supplied CAD data. This is also the first published work on studying the effects of randomising the regular lattices and the observed variations in strength. Both Mullen et al and Tsopanos et al used the software early in its development significantly aiding the testing and so progression of the computer tools.

7.3 Suggestions for Further Research

7.3.1 Feeding Back the Techniques Used on Horizontal Links

Links where successfully created at angles at low angles to the horizontal surface, but they where not ideal as they where more irregular than higher angle links, and more bulky. Techniques similar to those developed for the horizontal links could be attempted for the low angle links.

The success of the horizontal links have highlighted the benefits of reducing the volume of molten material while processing structures that are built significantly onto a free powder bed. Further investigation could establish whether this

technique could be extended to improving the surface finish on low angle and horizontal surfaces and solids.

7.3.2 Geometry Aware Slicing and Hatching Algorithms

Current slicing algorithms seem (the available programs are largely closed source, so the finer details are not clear) to only be concerned with the intersection of the CAD data and the slice plane that is currently being processed, and at most the previous layer in order to establish which areas on the current slice are overhanging. Research could be directed into creating the instructions for the laser and scanners based more closely to the 3D data, rather than on a slice and previous slice method. While this would increase computer resource requirements the advent of the availability of computers with multiple core and 8GB+ of RAM to the general public has left a great deal more resource available.

7.3.3 Development of Horizontal Link Scanning Techniques

Some of the horizontal links successfully formed, but they where often brittle. This may be due to letting the melt pool to cool too far, and the following exposure by the laser not re-melting much of the previous. This could result in a poor grain structure. Further investigation could focus on links created using the developed techniques and investigating if the microstructure of the links varies significantly with the angle from the horizontal. A possible route to improve the properties of the links may be to reduce the delay between pulses of the laser. By varying the delays the length of the melt could be adjusted so that the melt does not approach the Rayleigh instabilities, but improves grain structure.

The methods proposed by this research have demonstrated capabilities of the

machine beyond what was previously possible. However the productivity of the machine will be significantly reduced as there are additional delays being introduced. Currently the delay is simply one where the laser shuts off and the scanners don't move for a period of time. Modern scanners are capable of moving at very high speeds. The delay could be introduced by melting another point elsewhere, rather than doing nothing. This ordering would be best done as close to the hardware as possible as the precise capabilities of the scanners and laser will vary from system to system. As this will be difficult to achieve within a short time scale in a research environment it may be beneficial to optimise the jump speeds and re-ordering the points in the slice data to achieve the same effect, and so improve productivity.

7.3.4 Processing on a Free Powder Bed

Further research is suggested to find out more about the range of process parameters that could create this self-supporting effect. If the range is large enough and the structures reliable enough this could be of use to applications such as using SLM to generate large cells size sandwich panel cores. Over the course of collaborative research it has been observed that the stresses in the creation of the smaller cell size lattice and lattice-solid builds would be too great for this technique.

References

1: Stahl D.C., Gervasi V., Design and fabrication of components with optimized lattice microstructures, Pages: 838-844, Proceedings of the Solid Freeform Fabrication Symposium, 2004

2: Chiras S., Mumm D.R., Evans A.G., Wicks N., Hutchkinson J.W., Dharmasena K., Wadley H.N.G., Fichter S., The structural performance of near optimized truss core panels., International Journal of Solids and Structures, Pages: 4093-4115, Volume: 39, Pergamon, 2002

3: Gibson I., Shi D., Matereial properties and fabrication parameters in selective laser sintering process, Rapid Prototyping Journal, Pages: 129-136, Number: 4, Volume: 3, 1997

4: Banhart J., Manufacture, characterisation and application of cellular metals and metal foams, Progress in Materials Science, Pages: 559-632, Volume: 46, 2001

5: Wicks N., Hutchinson J.W., Optimal truss plates., International Journal of Solids and Structures, Pages: 5165-5183, Volume: 38, Pergamon, August, 2001

6: Evans A.G., Hutchinson J.W., Ashby M.F., Multifunctionality of cellular metal systems, Progress in Materials Science, Pages: 171-221, Volume: 43, 1999

7: Miyoshi T., Itoh M., Mukai T., Kanahashi H., Kohzu H., Tanabe S., Higashi K., Enhancement of energy absorption in a closed-cell aluminum by the modification of cellular structures., Scripta Materialia, Pages: 1055-1060, Number: 10, Volume: 41, 1999

8: Lu T.J., Hess A., Ashby M.F., Sound absorption in metallic foams., Journal of

Applied Physics, Pages: 7528-7533, Number: 11, Volume: 85, American Institute of Physics, June, 1999

9: Sypeck D.J., Cellular truss core sandwich structures, Applied Composite Materials, Pages: 229-246, Volume: 12, 2005

10: Tian J., Kim T., Lu T.J., Hodson H.P., Queheillalt D.T., Sypeck D.J., Wadley H.N.G., The effects pf topology upon fluid-flow and heat-transfer within cellular copper structures., Onternational Journal of Heat and Mass Transfer, Pages: 3171-3186, Volume: 47, 2004, Elsevier

11: Kudva J.N., Sanders B., Pinkerton-Florance J., Garcia E., The DARPA/AFRL/NASA smart wing program - final overview, Smart Structures and Materials 2002, Pages: 37-43, Volume: 4698, SPIE, 2002

12: Wicks N., Hutchinson J.W., Sandwich plates actuated by a kangome planar truss, Transactions of the ASME, Pages: 652-662, Volume: 71, ASME, September, 2004

13: Lin C.Y., Wirtz T., LaMarca F., Hollister S.J., Structural and mechanical evaluations of a topology optimized titanium interbody fusion cage fabricated by selective laser melting process, Wiley Interscience, 2006

14: Wohlers, T.;, Wohlers Report, Rapid Prototyping and Tooling, State of the Industry, Wohloers Associates, 2005, Wohlers Associates

15: Heinl, P.; Rottmair, A.; Korner, C.; Singer, R.;, Cellular Titanium by Selective Electron Beam Melting, Advanced Engineering Materials, Pages: 360-364, Number: 5, Volume: 9, 2007

16: Resch, M.; Kaplan, A.; Schoucker, D.;, Laser-assisted generating of three-

dimensional parts by the blown powder process, XIII International Symposium on Gas Flow and Chemical Lasers and High-Power Laser Conference, Volume: 4184, 2001, SPIE

17: Jeng J.Y.; Peng S.C.; Chou C.J., Metal Rapid Prototype Fabrication Using Selective Laser Cladding Technology, The International Journal of Advanced Manufacturing Technology, Pages: 681-687, Volume: 16, Number: 9, 2000

18: Tsopanos, S.; Mines, R.; McKown, S.; Shen, Y.; Cantwell, W.; Brooks, W.; Sutcliffe, C.;, The Influence of Processing Parameters on the Mechanical Properties of Selectively Laser Melted Stainless Steel Microlattice Structures, Journal of Manufacturing Science and Engineering, Volume: 132, August, 2010, ASME

19: Simone A.E., Gibson L.J., Aluminium foams produced by liquid-state processes., Acta Metallurica, Pages: 3109-3123, Number: 9, Volume: 46, Pergamon, 1998

20: Kenny L.D., Mechanical properties of particle stabilized aluminium foam., Materials Science Forum, Pages: 1883-1890, Volume: 217-222, 1996

21: Ashby M.F., Evans A., Fleck N.A., Gibson L.J., Hutchinson J.W., Wadley H.N.G., Metal Foams A Design Guide, Butterworth Heinemann, 2000, ISBN: 9780750672191

22: Miyoshi et al 2000, ALPORAS Aluminium foam: Production process, properties, and applications., Advanced Engineering Materials, Pages: 179-183, Number: 4, Volume: 4, 2000

23: Banhart J., Baumeister J., Webber M.,, Powder metallurgical technology for

the production of metal foam, Proceedings of the European Conference on Advanced PM Materials, 1995

24: Wadley H.N.G, Fleck N.A., Evans A.G., Composites Science and Technology, Pages: 2331-2343, Volume: 63, 2003

25: Brittain S.T., Sugimura Y., Schueller O.J.A., Evans A.G., Whitesides G.M., Fabrication and mechanical performance of a mesoscale space-filling truss system., Journal of Micro Electro Mechanical Systems, Pages: 113-120, Number: 1, Volume: 10, IEEE, March, 2001

26: Brandner J.J., Hansjosten E., Anurjew E., Pfleging W., Schubert K., Microstructure devices generation by selective laser melting, Proceeedings of SPIE, Volume: 6459, SPIE, 2007

27: Childs T.H.C., Hauser C., Badrossamay M., Selective laser sintering (melting) of stainless and tool steel powders, Part B Journal of Engineering Manufacture, Pages: 339-357, Volume: 219, Institution of Mechanical Engineers, 2005

28: Childs T.H.C., Hauser C., Raster scan selective laser melting of the surface layer of a tool steel powder bed, Part B Journal of Engineering Manufacture, Pages: 379-384, Volume: 219, Institution of Mechanical Engineers, 2005

29: Powers T.R., Goldstein R.E., Pearling and pinching: propagation of Rayleigh instabilities, Physical Review Letters, Pages: 2555-2558, Number: 13, Volume: 78, 1997

30: Powers T.R., Zhang D., Goldstein R.E., Stone H.A., Propagation of a topological transition: The Rayleigh instability, Physics of Fluids, Pages: 1052-1057, Number: 5, Volume: 10, American Institute of Physics, May, 1998

31: Grinfield M., On the Plateau-Rayleigh instability of truncated cylinder, Mechanics Research Communications, Pages: 613-616, Number: 6, Volume: 21, 1994

32: Amberg G., Do-Quang M., Thermocapillary convection and phase change in welding, International Journal of Numerical methods for heat & Fluid Flow, Number: 3/4, Volume: 18, 2008

33: Morgan R.H., Papworth A.J., Sutcliffe C., Fox P., O'Niell W., , High density net shape components by direct laser re-melting of single-phase powders, Journal of Materials Science, Pages: 3093-3100, Volume: 37, 2002

34: Shanping L.U., Hidetoshi F., Kiyoshi N., Effects of CO" Shielding gas additions and welding speed on GTA weld shape, Journal of Material Science, Pages: 2481-2485, Volume: 40, 2005

35: Fujii H., Sato T., Lu S., Nogi K., Development of an advanced A-TIG (AA-TIG) welding method by control of Marangoni convection, Materials Science and Engineering, Pages: 296-303, Volume: 495, 2008

36: Morgan R., Sutcliffe C.J., O'Niell W., Experimental investigation of nanosecond pulsed Nd:YAG laser re-melted pre-placed powder beds, Rapid Prototyping Journal, Pages: 159-172, Number: 3, Volume: 7, 2001

37: Niu H.J. Chang I.T.H., Liquid phase sintering of M3/2 high speed steel by selective laser sintering, Scripta Materialia, Pages: 67-72, Number: 1, Volume: 39, 1998

38: Rombouts m., Kruth J.P., Froyen L., Mercelis M., Fundementals of selective laser melting of alloyed steel powders

39: Wang, H; Rosen, D;, Computer-Aided Design Methods for Additive Fabrication of Truss Structures, 2005

40: Williams, C.; Mistree, F.; Rosen, D.;, Towards the design of a layer-based additive manufacturing process for the realization of metal parts of designed mesostructure, Proceedings of the Solid Freeform Fabrication Symposium, 2005

41: Schroeder W., Martin K., Lorensen B., Visualization Toolkit: An Object-Orientated Approach to 3D Graphics, edition: 4, Kitware, December, 2006, ISBN: 9781930934191

42: Rehme, O.; Emmemann, C., Rapid Manufacture of Lattice Structures with Selective Laser Melting, Proceedings of the SPIE, 2006

43: Brooks, W; Tsopanos, S; Stamp, R.; Sutcliffe, C.; Fox, P.; Todd, J.; , The Production of Open Cellular Lattice Structures Using Selective Laser Melting, Proceedings of the Sixth National Conference on Rapid Design, Prototyping and manufacture., 2005

44: Brooks, W.; Sutcliffe, C.; Cantwell, W.; Todd, J.; Mines, R.;, Rapid Design and Manufacture of Ultralight Cellular Materials, Proceedings of the Solid Freemform Fabrication Symposium 2005, 2005

45: Wang, H; Chen, Y; Rosen, D;, A Hybrid Geometric Modeling Method for Large Scale Conformal Cellular Structures, 2005

46: Mullen, L; Stamp, R.; Fox, P.; Jones, E.; Ngo, C.; Sutcliffe, C.;, Selective Laser Melting: A Unit Cell Approach for the Manufacture of Porous, Titanium, Bone In-Growth Constructs, Suitable for Orthopaedic Applications. II. Randomized Structures, Journal of Biomedical Research Part B: Applied Biomaterials, Pages: 178-188, Volume: 92B, July, 2009

47: Santorinaios, M; Brooks, W.; Sutcliffe, C.; Mines, R;, Crush behaviour of open cellular lattice structures manufactured using selective laser melting, High Performance Structures and Materials III, Pages: 481-490, Volume: 85, 2006, WIT, ISBN: 1743-3509

48: Mullen, L; Stamp, R.; Brooks, W.; Jones, E.; Sutcliffe, C.;, Selective Laser Melting: A Unit Cell Approach for the Manufacture of Porous, Titanium, Bone In-Growth Constructs, Suitable for Orthopaedic Applications, June, 2008

49: McKown, S; Mines, R.; Tsopanos, S.; , Finite Element Modelling of Micro Lattice Cores, Number: IRC/501/07, University of Liverpool, 2007, Impact Research Centre

50: Hasan, R.; Mines, R.; Shen, E.; Tsopanos, S.; Cantwell, W.; Brooks, W.; Sutcliffe, C.;, Comparison of the Drop Weight Impact Performance of Sandwich Panels with Aluminium Honeycomb and Titanium Alloy Micro Lattice Cores, Applied Mechanics and Materials, Pages: 413-418, Volume: 24-25, 2010

Index of Figures

Index of Tables